Composite Learning for Robust and Effective Dense Predictions

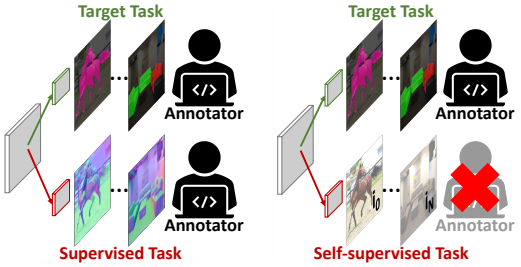
Menelaos Kanakis[1] Thomas E. Huang[1] David Bruggemann[1] Fisher Yu[1] Luc Van Gool[1,2]
[1]ETH Zürich [2]KU Leuven

Abstract

Multi-task learning promises better model generalization on a target task by jointly optimizing it with an auxiliary task. However, the current practice requires additional labeling efforts for the auxiliary task, while not guaranteeing better model performance. In this paper, we find that jointly training a dense prediction (target) task with a self-supervised (auxiliary) task can consistently improve the performance of the target task, while eliminating the need for labeling auxiliary tasks. We refer to this joint training as Composite Learning (CompL). Experiments of CompL on monocular depth estimation, semantic segmentation, and boundary detection show consistent performance improvements in fully and partially labeled datasets. Further analysis on depth estimation reveals that joint training with self-supervision outperforms most labeled auxiliary tasks. We also find that CompL can improve model robustness when the models are evaluated in new domains. These results demonstrate the benefits of self-supervision as an auxiliary task, and establish the design of novel task-specific self-supervised methods as a new axis of investigation for future multi-task learning research.

1. Introduction

Learning robust and generalizable feature representations have enabled the utilization of Convolutional Neural Networks (CNNs) on a wide range of tasks. This includes tasks that require efficient learning due to limited annotations. A commonly used paradigm to improve generalization of target tasks is Multi-Task Learning (MTL), the joint optimization of multiple tasks. MTL exploits domain information contained in the training signals of related tasks as an inductive bias in the learning process of the target task [8, 9]. The goal is to find joint representations that better explain the optimized tasks. MTL has demonstrated success in tasks such as instance segmentation [16] and depth estimation [12], amongst others. In reality, however, such performance improvements are not common when naively selecting the jointly optimized tasks [42]. To complicate things further, the relationship between tasks for MTL is also dependent on the learning setup, such as training set size and network capacity [56]. As a consequence, MTL practitioners are forced to iterate through various candidate task combinations in search of a synergetic setting. This empirical process is arduous and expensive since annotations are required *a priori* for each candidate task.



(a) Multi-Task Learning (b) Composite Learning (ours)

Figure 1. The generalization of target tasks can be improved by jointly optimizing with a related auxiliary task. (a) In traditional multi-task learning, one uses labeled auxiliary tasks that require manual annotation efforts. (b) In this paper, we show that jointly training a dense task with a self-supervised task can consistently improve the performance, while eliminating the need for additional labeling efforts.

In this paper, we find that the joint optimization of a dense prediction (target) task with a self-supervised (auxiliary) task improves the performance on the target task, outperforming traditional MTL practices. We refer to this joint training as Composite Learning (CompL), inspired by material science where two materials are merged to form a new one with enhanced properties. The benefits and intuition of CompL resemble those of traditional MTL, however, CompL exploits the label-free supervision of self-supervised methods. This facilitates faster iterations through different task combinations, and eliminates manual labeling effort for auxiliary tasks from the process.

We provide thorough evaluations of CompL on three dense prediction target tasks with different model structures, combined with three self-supervised auxiliary tasks. The target tasks include depth estimation, semantic segmentation, and boundary detection, while self-supervised tasks include rotations, MoCo, and DenseCL. We find that jointly optimizing with self-supervised auxiliary tasks consistently

outperforms ImageNet-pretrained baselines. The benefits of CompL are most pronounced in low-data regimes, where the importance of inductive biases increases [5]. We also find that jointly optimizing monocular depth estimation with a self-supervised objective can outperform most labeled auxiliary tasks. CompL can additionally improve semantic segmentation and boundary detection model robustness, when evaluated on new domains. Our experiments demonstrate the promise of self-supervision as an auxiliary task. We envision these findings will establish the design of novel task-specific self-supervised methods as a new axis of investigation for future multi-task learning research.

2. Related Work

Multi-Task Learning (MTL) MTL aims to enhance performance and robustness of a predictor by jointly optimizing a shared representation between several tasks [8]. This is accomplished by exploiting the domain-specific information contained in the training signal of one task (e.g., semantic segmentation), to more informatively select hypotheses for other tasks (e.g., depth), and vice versa [52, 7]. For example, pixels of class "*sky*" will always have a larger depth that those of class "*car*" [54]. If non-related tasks are combined, however, the overall performance degrades. This is referred to as task-interference and has been well documented in the literature [47, 40]. However, no measurement of task relations can tell us whether performance gain can be achieved without training the final models. Although several works have shown that while MTL can improve performance, it requires an exhaustive manual search of task interactions [56], and labeled datasets with many tasks. In this work we also jointly optimize a network on multiple tasks, but we instead evaluate the efficacy of self-supervision as an auxiliary task. This enables the use of joint training in any dataset and eliminates expensive annotation efforts that do not guarantee performance gains. To further improve performance of a target task, [37, 29, 4, 24] designed specialised architectures for a predefined set of tasks. These architectures do not generalize to other tasks. On the other end, [45] aim to learn a sub-class labelling problem as an auxiliary task, i.e. for class dog learn the breed subclass, however the notion of subclass does not generalize to dense tasks like depth estimation. Instead, we conduct a systematic investigation using a common pipeline, applicable to any dense target task. This enables the easy switching of different supervised target tasks or auxiliary self-supervised tasks, without requiring any architectural changes, enabling the wider reach of joint training across tasks and datasets.

Transfer learning Given a large labeled dataset, neural networks can optimize for any task, whether image-level [43], or dense [32]. In practice, however, large datasets can be prohibitively expensive to acquire, giving rise to the transfer learning paradigm. The most prominent example of transfer learning is the fine-tuning of an ImageNet [17] pre-trained model on target tasks such as semantic segmentation [46], or monocular depth estimation [22]. However, ImageNet models do not always provide the best representations for all downstream tasks, raising interest in finding task relationships for better transfer capabilities [68]. In this work we are not interested in learning better pre-trained networks for knowledge transfer. Rather, we start from strong transfer learning baselines and improve generalization by jointly optimizing the target and auxiliary tasks.

Self-supervised learning Learning representations that can effectively transfer to downstream tasks, coupled with the cost associated with the acquisition of large labeled datasets, has given rise to self-supervised methods. These methods can learn representations through explicit supervision on pre-text tasks [19, 27], or through contrastive methods [13, 31]. Commonly, self-supervised methods aim to optimize a given architecture, yielding better pre-training models for fine-tuning on the target task [19, 27, 13, 31, 25, 62, 49, 44]. We instead utilized such pre-trained models as a starting point and fine-tune on both the target and self-superivsed auxiliary tasks jointly, rather than just the target task, to further improve performance and robustness. More recently, supervised tasks have been used in conjunction with self-supervised techniques by exploiting the labels to guide contrastive learning. This can be seen as a form of sampling guidance and has been utilized in classification [41], semantic segmentation [60], and tracking [51]. These methods differ from our work as they require target task labels to optimize the self-supervised objective, while our self-supervised objectives are independent of the target labels and can be applied on any set of images. Instead, [18] jointly train a model for classification and rotation, but utilize the rotation performance at test time as a proxy to the classification performance. The goal of this work is instead to improve the target task's performance and robustness. More closely to our work, [26] and [69] jointly train classification and self-supervised objectives under a semi-supervised training protocol. We also perform joint training with a self-supervised task, however, we follow a more general MTL methodology, and investigate whether self-supervised tasks can provide inductive bias to dense tasks.

Robustness Robust predictors are important to ensure their performance under various conditions during deployment. Recent works have focused on improving different aspects of robustness, such as image corruption [35], adversarial samples [70], and domain shifts [65]. More related to our work, [36] jointly train classification and self-supervised rotation, demonstrating that the strong regularization of the rotations improves model robustness to adversarial examples, and label or input corruptions. [61] similarly used joint training but employed both image and video-level self-

supervised tasks and found them to improve the model's robustness to domain shifts for object detection. We also evaluate the effect of joint training on robustness to unseen datasets, but focus on dense prediction tasks.

3. Composite Learning

In this section, we introduce and motivate Composite Learning (CompL). Specifically, Sec. 3.1 formalizes the problem setting, Sec. 3.2 describes the self-supervised methods investigated, and Sec. 3.3 lists the network structure choices in our study.

3.1. Joint Learning with Supervised and Self-Supervised Tasks

Multi-task learning may improve the model robustness and generalizability. We aim to investigate the efficacy of joint training with self-supervision on dense prediction tasks as the targets. The shared representation between the target task t and an auxiliary task a may be more effective than training on t alone.

In the traditional MTL setup, the label sets Y_t, and Y_a, are manually labeled. In contrast, the auxiliary labels Y_a in CompL are implicitly created in the self-supervised task. Formally, CompL aims to produce the two predictive functions $f_t(\theta_s, \theta_t) : \mathcal{X}_t \rightarrow \mathcal{Y}_t$ and $f_a(\theta_s, \theta_a) : \mathcal{X}_a \rightarrow \mathcal{Y}_a$, where f_t and f_a share parameters θ_s and have disjoint parameters $\theta_{\{t,a\}}$. During inference we are only interested in f_t, however, we hypothesize that we can learn a more effective parameterization through the above weight sharing scheme. In our investigation, f_t and f_a are trained jointly using samples (X_t, y_t) and (X_a, y_a).

The overall optimization objective therefore becomes

$$\min_{\theta_s, \theta_t, \theta_a} \mathcal{L}^t((X_t, y_t); \theta_s, \theta_t) + \lambda \mathcal{L}^a((X_a, y_a); \theta_s, \theta_a), \quad (1)$$

where $\mathcal{L}^t$ and $\mathcal{L}^a$ are the losses for the supervised and self-supervised tasks respectively, and λ is a scaling factor controlling the magnitude and importance of the self-supervised task.

The experiments in this paper use the same dataset for both the target and auxiliary tasks. We additionally train our models using different-sized subsets (X'_t, y'_t) for the target task, where $X'_t \subseteq X_t = X_a$. However, the above is not a necessary condition for CompL, meaning the self-supervised task could be trained on an independent dataset.

Training method We jointly optimize two objectives. We construct a minibatch by sampling at random independently from the two training sets. For simplicity, we sample an identical number of images from each training set. The input images X_t and X_a are treated independently. This enables us to apply task/method-specific augmentations to each task input without causing task conflicts. We apply the baseline augmentations to X_t, ensuring a fair comparison with our single-task baselines. X_a used for self-supervised training is instead processed with the proposed task-specific augmentations for each method investigated. These augmentations include Gaussian blur and rotation. They can significantly degrade performance for dense tasks if applied on the target task, but they are important for self-supervision. Therefore, by using distinct augmentations on two tasks, we can minimize performance degradation brought by training the auxiliary tasks.

3.2. Self-Supervised Methods in Our Study

Rotation (Rot) [27] proposed to utilize 2-dimensional rotations on the input images to learn feature representations. Specifically, they optimize a classification model to predict the rotation angles, equally spaced in $[0°, 360°)$. Joint optimization with self-supervised rotation has demonstrated success in semi-supervised image classification [26, 69], and enhanced robustness to input/output corruptions [36], making it a prime candidate for further investigation in a dense prediction setting.

Global contrastive Global contrastive methods treat every image as its own class, while artificially creating novel instances of said class through random data augmentations. In this work, we evaluate contrastive methods using Momentum Contrast (MoCo) [31], and specifically MoCo v2 [14]. These methods formulate contrastive learning as dictionary look-up, enabling for the construction of a large and consistent dictionary of size $|Z|$ without the need for large batch sizes, a common challenge amongst dense prediction tasks [11]. MoCo is optimized using InfoNCE [50], a contrastive loss function defined as

$$\mathcal{L} = -\log \frac{\exp(z^+/\tau)}{\sum_{z \in Z} \exp(z/\tau)}. \quad (2)$$

InfoNCE is a softmax-based classifier that optimizes for distinguishing the positive representation z^+ from the $|Z| - 1$ negative representations. The temperature τ is used to control the smoothness of the probability distribution, with higher values resulting in softer distributions.

Local contrastive In dense predictions tasks, we desire a fine-grained pixel wise prediction rather than a global one. As such, we further investigate the difference between global contrastive MoCo v2 [14], and its variant DenseCL [62], that includes an additional contrastive loss acting on local representations.

3.3. Network Structures

Dense prediction networks are initially pre-trained on classification, and then modified according to the downstream task of interest, e.g., by introducing dilations [66]. In our investigation, we jointly optimize heterogeneous tasks

Table 1. Monocular depth estimation performance in RMSE on NYUD-v2. '→' denote transfer learning methods, while '+' denote joint training (CompL). Initialization with DenseCL coupled with DenseCL joint training outperforms all other methods.

Model	Labeled Data				
	5%	10%	20%	50%	100%
Depth	0.8871	0.8120	0.7471	0.6655	0.6223
Rot → Depth	1.0830	1.0120	0.9114	0.8322	0.7822
MoCo → Depth	0.8758	0.7708	0.7113	0.6311	0.5890
DenseCL → Depth	0.8736	0.7726	0.7152	0.6321	0.5982
Depth + Rot	0.8762	0.8071	0.7298	0.6460	0.6107
Depth + MoCo	0.8501	0.7955	0.7206	0.6434	0.6000
Depth + DenseCL	0.8479	0.7866	0.7131	0.6420	0.5990
MoCo → MoCo + Depth	0.8614	0.7732	0.7008	0.6220	0.5773
DenseCL → DenseCL + Depth	**0.8468**	**0.7641**	**0.6989**	**0.6157**	**0.5690**

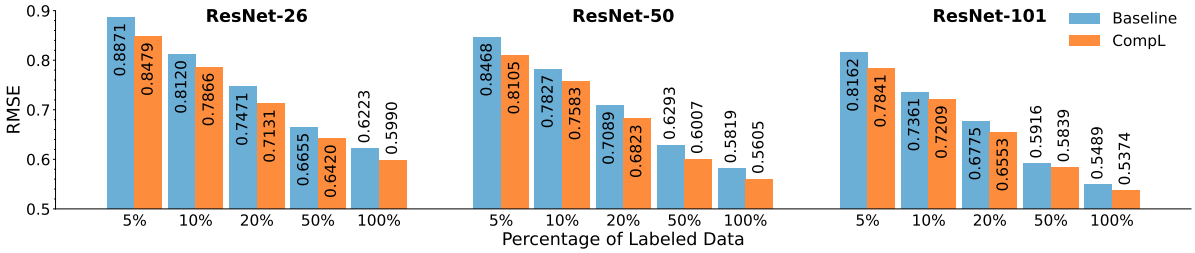


Figure 2. Monocular depth estimation performance in RMSE on different ResNet encoders. Use of CompL (orange) denotes the addition of the best performing self-supervised objective (DenseCL). CompL consistently outperforms the baselines in all experiments.

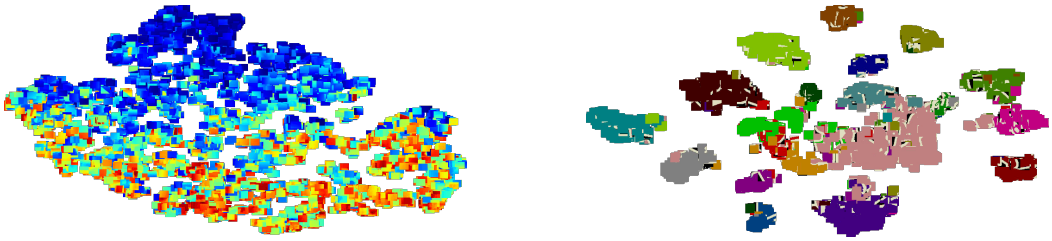
such as a dense prediction task and image rotations. Therefore, our networks call for special structure considerations. This section presents the details.

Dense prediction networks Common dense prediction networks use an encoder-decoder structure [53, 3], maintain a constant resolution past a certain network depth [67], or even utilize both high and low representation resolutions in multiple layers of the network [59]. Due to the large differences among networks, we opt to treat the entire network as a single unit, and only utilize the last feature representation of the networks for the task-specific predictions. In other words, we branch out at the last layer and employ a single task-specific module for the predictions. This ensures that our findings do not depend on network structures, and it is easy to generalize to new network designs.

We perform our experiments on DeepLabv3+ [11] based on ResNets [33]. The networks demonstrated competitive performance on a large number of dense prediction tasks, such as semantic segmentation, and depth estimation and has been used extensively when jointly learning multiple tasks [47, 6]. Our investigation is primarily on the smaller ResNet-26 architecture for easy comparison with existing MTL results. As it is a common practice in dense prediction tasks, we initialize the ResNet encoder with ImageNet pretrained weights, unless stated otherwise.

Task-specific heads The final representation of the dense prediction networks is utilized in two task-specific modules. The first module, consisting of a 1×1 convolutional layer, generates the predictions of the supervised task, with the output dimension being task dependent, such as the number of classes. The second prediction head is specific for self-supervised tasks. Unlike the supervised prediction head, the self-supervised prediction head is utilized only during network optimization, and is discarded at test time. The features for Rot and MoCo are first pooled with a global average pooling layer. Rot is then processed by an fully connected layer with output dimensions equal to 4, number of potential rotations, while MoCo is processed by 2-layer MLP head with output dimensions equal to 128, feature embedding dimension. DenseCL, on the other hand, generates two outputs. The first one is identical to MoCo for the global representation, while for the second representation, the initial dense features are pooled to a smaller grid size, and then processed with two 1×1 convolutional layer to get the local feature representations.

Normalization Large CNNs are often challenging to train, and thus utilize Batch Normalization (BN) to accelerate training [38]. In self-supervised training, BNs often degrade performance due to intra-batch knowledge transfer among samples. Workarounds include shuffling BNs [31, 14], using significantly larger batch sizes [13], or even replacing BNs altogether [34]. To ensure BNs will not affect our study, and findings can be attributed to the jointly trained tasks, we replace BNs with group normalization (GN) [63]. We chose GN as it yielded the best performance when trained on ImageNet [63]. However, other normalization layers that are not affected by batch statistics can also be utilized, such as layer [2] and instance [57] normalizations.



(a) Monocular depth estimation. (b) Semantic segmentation.

Figure 3. t-SNE visualization of the DenseCL local representations. The representations are depicted using their ground-truth maps. Specifically, (a) depth values for monocular depth estimation and (b) semantic patches for semantic segmentation. The local representations adapt to the target task, i.e., (a) smooth depth variation for the regression task while (b) clusters are formed for the classification task.

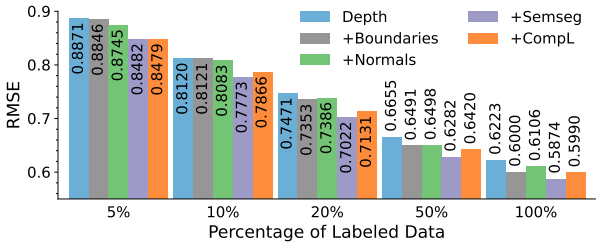


Figure 4. Monocular depth estimation performance in RMSE on NYUD-v2 when trained with additional auxiliary tasks. CompL can improve depth more than training with boundary or normal predictions. Semantic segmentation can improve the depth prediction more, but it requires expensive manual annotations.

4. Experiments

In this section we investigate the effects of jointly training dense predictions and self-supervised tasks. To systematically assess the effect of joint learning in label-deficient cases, we use different-sized subsets (X'_T, y'_t) of the full target task data (X_T, y_t), i.e., $(X'_T, y'_t) \subseteq (X_T, y_t)$. To ensure consistent contribution from the auxiliary task, we always use the full data split (X_A, y_a) for the self-supervised task. The supplementary material includes additional experiments using the same subsets for both tasks.

Implementation details We sample 8 images at random from each of the target and auxiliary training sets. We apply the baseline augmentations to target samples, namely, random horizontal flipping, random image scaling in the range [0.5, 2.0] in 0.25 increments, and then crop or pad the image to ensure a consistent size. The auxiliary loss is scaled by λ. We found 0.2 works best for MoCo and DenseCL, while 0.05 for Rot. The model is optimized using stochastic gradient decent with momentum 0.9, weight decay 0.0001, and the "poly" learning rate schedule [10].

4.1. Monocular Depth Estimation

We first evaluate CompL on monocular depth estimation. Monocular depth estimation is a widely used dense prediction task, and is typically casted as a regression problem.

Experimental protocol Monocular depth estimation is explored on NYUD-v2 [55], comprised of 795 train and 654 test images from indoor scenes, and evaluated using the root mean squared error (RMSE) metric. All models are trained for 20k iterations, corresponding to 200 epochs of the fully labeled dataset, with an input image size of 425×560, and are optimized with the $\mathcal{L}_1$ loss.

Joint optimization Table 1 presents the performance of the single-task baseline, "Depth", and the models trained jointly with different self-supervised tasks, "Depth + *Task name*". We find that joint training with any self-supervised task consistently improves the performance of the target task, even in the fully labeled dataset. In particular, joint training with self-supervision yields the biggest performance improvements on the lower labeled percentages, where the importance of inductive bias increases [5]. These findings are consistent also when utilizing stronger ResNet encoders, as depicted in Fig. 2 for the best performing self-supervised DenseCL method.

DenseCL contrasts both local and global representations, yielding richer representations for dense task pre-training, as compared to the image-level self-supervised tasks. We find this to also be the case in our joint-training setup, where richer local representations help guide the optimization of depth. To better understand the benefit of utilizing DenseCL for joint training with depth, we visualize the representations in Fig. 3a using a t-SNE plot [58]. Specifically, we depict the latent representations of DenseCL using their corresponding ground-truth depth measurements. The depth values smoothly transition from larger distances (in red) to smaller distances (in blue). This indicates that the DenseCL objective, which is discriminative by construction, promotes a smooth variation in the representations when combined with a regression target objective.

Traditional MTL In order to determine how CompL compares to traditional MTL, we evaluate and compare the effect of using labeled auxiliary tasks. Specifically, we investigate the effect of the remaining three tasks of NYUD-v2, that is, boundaries, normals, and semantic segmentation, in Fig. 4. For fair comparisons to CompL, the auxiliary tasks also use the entire dataset. CompL consistently outperforms the use of labeled boundaries and normals as

Table 2. Semantic segmentation performance in mIoU on the PASCAL VOC dataset. '→' denote transfer learning methods, while '+' denote joint training (CompL). Joint training with DenseCL significantly outperforms the "Semseg" baselines.

Model	Labeled Data						
	1%	2%	5%	10%	20%	50%	100%
Semseg	30.82	37.66	49.95	55.17	61.30	67.38	70.42
Rot → Semseg	10.35	12.43	18.29	24.71	29.21	35.43	39.46
MoCo → Semseg	31.55	37.55	48.60	53.27	58.74	64.04	68.09
DenseCL → Semseg	34.89	39.72	50.96	55.60	61.13	65.71	69.56
Semseg + Rot	28.75	36.81	50.46	56.21	62.17	67.96	70.52
Semseg + MoCo	32.90	40.31	52.18	56.50	62.49	68.40	71.15
Semseg + DenseCL	33.51	40.91	52.76	**57.33**	**63.22**	**68.81**	**71.16**
DenseCL → Semseg + DenseCL	**36.32**	**41.24**	**52.94**	56.87	62.71	65.89	69.81

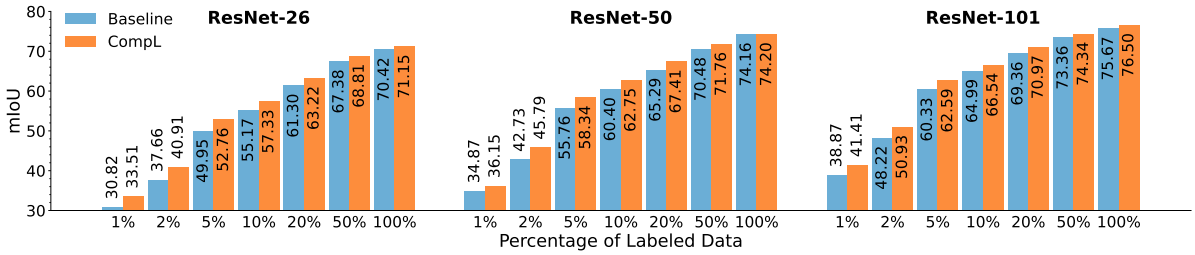


Figure 5. Semantic segmentation performance in mIoU on different ResNet encoders. Use of CompL (orange) denotes the addition of the best performing self-supervised objective (DenseCL). CompL consistently outperforms the baselines in all experiments.

auxiliary tasks. This is particularly pronounced in the lower data splits where the contribution of CompL becomes more prominent, while boundaries and normals contribute less. Surface normals, derivatives of depth maps, could be expected to boost depth prediction due to their close relationship. However, we find it to help only marginally. On the other hand, joint training with semantic segmentation consistently improves the baseline performance, which aligns with findings in the previous works [12, 28, 39]. These results exemplify the importance of an arduous iteration process in search of a synergistic auxiliary task, where knowledge of label interactions are not necessarily helpful. This process is further complicated when additional auxiliary task annotations are needed. Therefore, eliminating manual labeling from auxiliary tasks opens up a new axis of investigation for the future of multi-task learning research as it can enable faster iterations in task interaction research.

Transfer learning The experiments have so far shown that joint training with self-supervision can enhance performance, and in most cases outperforms traditional MTL practices. Notably, outperforming the baselines even when all models are initialized with ImageNet pre-trained weights, a strong transfer learning baseline. However, is ImageNet pre-training the best initialization for Depth, and how does it compare to self-supervised pre-training? In Table 1 we repeat the baseline experiments starting from self-supervised pre-training, ("*Initial task* → Depth"). In depth estimation, the contrastive methods gain the advantage and outperform the joint training methods. However, our proposed method is not limited by the initialization used. We find that initialization with MoCo or DenseCL weights coupled with joint training ("*Initial task* → *Initial task* + Depth") can increase the performance even further, giving the best performing models.

4.2. Semantic Segmentation

We additionally evaluate semantic segmentation. Semantic segmentation is representative for discrete labeling dense predictions.

Experimental protocol Semantic segmentation (Semseg) experiments are conducted on PASCAL VOC 2012 [21], and specifically the augmented version (aug.) from [30], that provides 10,582 train and 1,449 test images. We evaluate performance in terms of mean Intersection-over-Union (mIoU) across the classes. All models are trained for 80k iterations, accounting for 60 epochs of the fully labeled dataset, and are optimized with the cross-entropy loss with image input size of 512×512.

Joint optimization Table 2 present the performance of the single-task baseline and the models trained jointly with different self-supervised tasks. In contrast to findings from classification literature [26, 69], joint training with Rot minimally affects the performance in most cases, with lower labeled percentages even incurring a performance degradation. On the other hand, the contrastive methods increase performance on all labeled splits, with lower labeled percentages incurring the biggest performance improvement. These findings are once again consistent when utilizing stronger ResNet encoders, as depicted in Fig. 5 for the best performing self-supervised method DenseCL. Similar

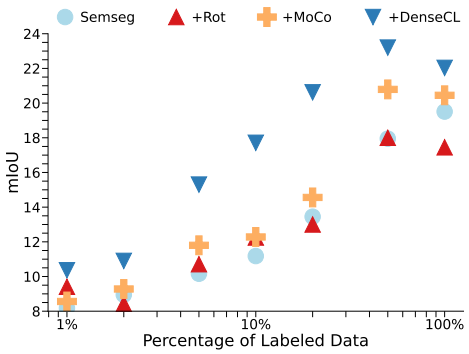


Figure 6. Semantic segmentation performance in mIoU trained on PASCAL VOC and evaluated on BDD100K. The local contrastive loss of DenseCL provides significant robustness improvements.

to depth, we further visualize in Fig. 3b the latent representations contrasted by DenseCL, and depict them with their ground-truth semantic maps. Unlike in depth regression, where the representations were smooth due to the continuous nature of the problem, the DenseCL representations for semantic segmentation form clusters given the discriminative nature of semantic segmentation.

Robustness to zero-shot dataset transfer So far we have only evaluated on the same distribution as that used for training, however, distribution shifts during deployment are common. We therefore investigate the generalization capabilities to new and unseen datasets. We evaluate the zero-shot capabilities of the models on the challenging BDD100K [65] dataset in Fig. 6, a diverse driving dataset. The test frames from BDD100K are therefore significantly different to those observed during training, making zero-shot transfer particularly interesting due to the large domain shift. We report the mIoU with respect to the shared classes between the two datasets. Please refer to the supplementary for the table of the BDD100K experiments.

We find that Rot often performs worse than the baseline model. This yields dissimilar findings to classification [36] that observed increased robustness, attributed to the strong regularization induced by the joint training. For Semseg, such regularizations degrade the fine-grained precision required. Joint training with DenseCL significantly outperforms all other self-supervised methods. While MoCo was comparable to DenseCL on VOC (Table 2), we find that local contrastive plays a big role in improving robustness. Interestingly, when using 100% of the data points, performance on all methods utilizing self-supervision is lower than when using 50% of the labels. We conjecture that, using the fully labeled split decreases the influence of self-supervision, making the model more prone to overfit to the training dataset and loose generalizability.

Transfer learning Table 2 additionally reports the baseline experiments starting from self-supervised pre-training (indicated by "*Initial task* → Semseg"), or additionally optimized with the best performing DenseCL method, as in the Depth experiments. Joint training with self-supervision consistently outperforms the sequential training counterpart, and in the majority of the cases by a significant margin. In other words, CompL consistently reports performance gains when initializing with either ImageNet or DenseCL.

Table 3. Boundary detection performance in ODS F-score on the BSDS500 dataset. '→' denote transfer learning methods, while '+' denotes joint training. Performance improvements are marginal, in contrast to the findings for other target tasks.

Model	Labeled Data			
	10%	20%	50%	100%
Boundaries	71.10	73.50	75.90	76.80
Rot → Boundaries	60.20	62.80	66.00	67.70
MoCo → Boundaries	71.00	73.40	75.60	76.40
DenseCL → Boundaries	68.90	71.70	75.40	75.90
Boundaries + Semseg	70.60	73.30	75.60	**76.90**
Boundaries + Rot	69.70	73.00	75.70	76.60
Boundaries + MoCo	**71.30**	73.80	**76.20**	**76.90**
Boundaries + DenseCL	**71.30**	**73.90**	76.00	76.20

4.3. Boundary Detection

Boundary detection is another common dense prediction tasks. Unlike depth prediction and semantic segmentation, the target boundary pixels only account for a small percentage of the overall pixels. We find that CompL significantly improves the model robustness for boundary detection.

Experimental protocol We study boundary detection on the BSDS500 [1] dataset, consisting of 300 train and 200 test images. Since the ground truth labels of BSDS500 are provided by multiple annotators, we follow the approach of [64] and only count a pixel as positive if it was annotated as positive by at least three annotators. Performance is evaluated using the Optimal-Dataset-Scale F-measure (ODS F-score) [48]. All models are trained for 10k iterations on input images of size 481×481. Following [64], we use a cross-entropy loss with a weight of 0.95 for the positive and 0.05 for the negative pixels.

Joint optimization Table 3 presents the performance of the single-task baseline and the models trained jointly with different self-supervised tasks. Compared to the previous two tasks, boundary detection is marginally improved by CompL. Since convolutional networks are biased towards recognising texture rather than shape [23], we hypothesize that the supervisory signal of contrastive learning interferes with the learning of edge / shape filters essential for boundary detection. To investigate this hypothesis further, we jointly train boundary detection with a labeled high-level semantic task. Specifically, we jointly train boundary detection with the ground-truth foreground-background segmentation maps for BSDS500 [1] from [20]. As seen in Table 3, the incorporation of semantic information once again does

Table 4. Performance of a multi-task model for monocular depth estimation in RMSE and semantic segmentation in mIoU on NYUD-v2. '+' denote joint training. The multi-task model combined with CompL yields consistent improvements in both tasks.

Model	Depth Labeled Data ↓					Semseg Labeled Data ↑				
	5%	10%	20%	50%	100%	5%	10%	20%	50%	100%
Depth + Semseg	0.997	0.904	0.794	0.665	0.606	10.46	14.99	19.41	26.24	31.66
Depth + Semseg + DenseCL	**0.902**	**0.806**	**0.744**	**0.641**	**0.590**	**10.72**	**15.29**	**20.08**	**28.18**	**33.48**

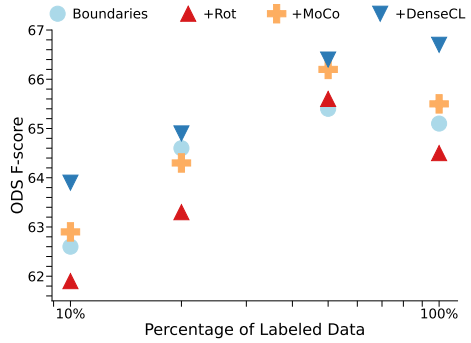


Figure 7. Boundary detection performance in ODS F-score trained on BSDS and evaluated on NYUD. The additional local contrast of DenseCL increases robustness to zero-shot dataset transfer.

not enhance the single-task performance of boundaries, and even slightly degrades at lower percentage splits.

While CompL yielded performance improvements for monocular depth estimation and semantic segmentation as target tasks, boundary estimation does not observe the same benefits. This further demonstrates the complexity of identifying a universal auxiliary task for all target tasks. Instead, it demonstrates the importance of co-designed self-supervised tasks alongside the downstream task.

Robustness to zero-shot dataset transfer We evaluate the zero-shot dataset transfer capabilities of the BSDS500 [1] models from Table 3 on NYUD-v2 [55]. Interestingly, even though CompL did not significantly improve the performance in Table 3, we find that the robustness experiments in Fig. 7 paint a different picture. While MoCo often outperformed DenseCL in Table 3, and most methods perform comparatively to the baseline, the additional local constrast of DenseCL significantly improves the robustness experiments. This can be seen from DenseCL consistently outperforming the baseline, as well as all other methods.

Transfer learning Table 3 also reports the performance of the boundary detection transfer learning experiments. All three transfer learning approaches fare worse than ImageNet initialization, corroborating our hypothesis that boundary detection requires representations which are fairly unrelated to the features learned through self-supervision.

4.4. Multi-Task Model (Semseg and Depth)

Both semantic segmentation (Semseg) and monocular depth estimation (Depth) observed improvements when trained under CompL. In this section, we further investigate the applicability of CompL on MTL models optimized jointly for Depth and Semseg (Depth + Semseg).

Experimental protocol We explore joint training on NYUD-v2 [55], which provides ground-truth labels for both tasks. We maintain the exact same hyperparameters as the models in Sec. 4.1, however, we expect an explicit search could yield additional improvements. No additional task-specific scaling of the losses is used, following [47]. For self-supervised tasks, we only evaluate DenseCL [62], as it performed the best for both tasks independently.

Joint optimization Table 4 presents the performance of the baseline multi-task model (Depth + Semseg) and the model trained jointly with DenseCL (Depth + Semseg + DenseCL). As in the single-task settings, training under CompL enhances the performance of both Semseg and Depth. Specifically, we again observe a performance gain in every labeled percentage. This demonstrates that, even in the traditional multi-task setting, the additional use of CompL has the potential of yielding further performance gains. In the current setting, Depth observes a noticeable gain over Semseg in low data regimes. This can be attributed to the DenseCL hyperparameters being optimized directly for the improvement of Depth. More advanced loss balancing schemes [15] could yield a redistribution of the performance gains, however, such investigation is beyond the scope of our work.

5. Conclusion

In this paper we introduced CompL, a method that exploits the inductive bias provided by a self-supervised task to enhance the performance of a target task. CompL exploits the label-free supervision of self-supervised methods, facilitating faster iterations through different task combinations. We show consistent performance improvements in fully and partially labeled datasets for both semantic segmentation and monocular depth estimation. While our method eliminated the need for labeling the auxiliary task, it commonly outperforms the traditional MTL with labeled auxiliary tasks on monocular depth estimation. Additionally, the semantic segmentation models trained under CompL yield better robustness on zero-shot cross dataset transfer. We envision our contribution to spark interest in the explicit design of self-supervised tasks for their use in joint training, opening up a new axis of investigation for future multi-task learning research.

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2010.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 39(12):2481–2495, 2017.

[4] Jongbeom Baek, Gyeongnyeon Kim, and Seungryong Kim. Semi-supervised learning with mutual distillation for monocular depth estimation. *arXiv preprint arXiv:2203.09737*, 2022.

[5] Jonathan Baxter. A model of inductive bias learning. *JAIR*, 12:149–198, 2000.

[6] David Bruggemann, Menelaos Kanakis, Stamatios Georgoulis, and Luc Van Gool. Automated search for resource-efficient branched multi-task networks. In *BMVC*, 2020.

[7] David Bruggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction. In *ICCV*, 2021.

[8] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[9] Rich Caruana. A dozen tricks with multitask learning. In *Neural networks: tricks of the trade*, pages 165–191. Springer, 1998.

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4):834–848, 2017.

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[12] Po-Yi Chen, Alexander H Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *CVPR*, 2019.

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[14] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[15] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 2018.

[16] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[18] Weijian Deng, Stephen Gould, and Liang Zheng. What does rotation prediction tell us about classifier accuracy under varying testing environments? In *ICML*, 2021.

[19] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[20] Ian Endres and Derek Hoiem. Category independent object proposals. In *ECCV*, 2010.

[21] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[22] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.

[23] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2018.

[24] Mariana-Iuliana Georgescu, Antonio Barbalau, Radu Tudor Ionescu, Fahad Shahbaz Khan, Marius Popescu, and Mubarak Shah. Anomaly detection in video via self-supervised and multi-task learning. In *CVPR*, 2021.

[25] Golnaz Ghiasi, Barret Zoph, Ekin D Cubuk, Quoc V Le, and Tsung-Yi Lin. Multi-task self-training for learning general representations. In *ICCV*, 2021.

[26] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *ICCV*, 2019.

[27] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ICLR*, 2018.

[28] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *ICLR*, 2020.

[29] Vitor Guizilini, Jie Li, Rares Ambrus, Sudeep Pillai, and Adrien Gaidon. Robust semi-supervised monocular depth estimation with reprojected distances. In *CoRL*, 2020.

[30] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[31] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[32] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *ICCV*, 2019.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[34] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 2020.

[35] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

[36] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, 2019.

[37] Lukas Hoyer, Dengxin Dai, Yuhua Chen, Adrian Koring, Suman Saha, and Luc Van Gool. Three ways to improve semantic segmentation with self-supervised depth estimation. In *CVPR*, 2021.

[38] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[39] Jianbo Jiao, Ying Cao, Yibing Song, and Rynson Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *ECCV*, 2018.

[40] Menelaos Kanakis, David Bruggemann, Suman Saha, Stamatios Georgoulis, Anton Obukhov, and Luc Van Gool. Reparameterizing convolutions for incremental multi-task learning without task interference. In *ECCV*, 2020.

[41] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

[42] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017.

[43] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[44] Zhaowen Li, Yousong Zhu, Fan Yang, Wei Li, Chaoyang Zhao, Yingying Chen, Zhiyang Chen, Jiahao Xie, Liwei Wu, Rui Zhao, et al. Univip: A unified framework for self-supervised visual pre-training. In *CVPR*, 2022.

[45] Shikun Liu, Andrew Davison, and Edward Johns. Self-supervised generalisation with meta auxiliary learning. *NeurIPS*, 2019.

[46] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[47] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *CVPR*, 2019.

[48] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *TPAMI*, 26(5):530–549, 2004.

[49] Alejandro Newell and Jia Deng. How useful is self-supervised pretraining for visual tasks? In *CVPR*, 2020.

[50] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[51] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, 2021.

[52] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *TPAMI*, 41(1):121–135, 2017.

[53] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[54] Suman Saha, Anton Obukhov, Danda Pani Paudel, Menelaos Kanakis, Yuhua Chen, Stamatios Georgoulis, and Luc Van Gool. Learning to relate depth and semantics for unsupervised domain adaptation. In *CVPR*, 2021.

[55] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[56] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *ICML*, 2020.

[57] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[58] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008.

[59] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2020.

[60] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *ICCV*, 2021.

[61] Xin Wang, Thomas E Huang, Benlin Liu, Fisher Yu, Xiaolong Wang, Joseph E Gonzalez, and Trevor Darrell. Robust object detection via instance-level temporal cycle confusion. In *ICCV*, 2021.

[62] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, 2021.

[63] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

[64] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, 2015.

[65] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020.

[66] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[67] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, 2017.

[68] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018.

[69] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *ICCV*, 2019.

[70] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.