Self-Supervised Learning in Multi-Task Graphs through Iterative Consensus Shift

Emanuela Haller*[1,2]
haller.emanuela@gmail.com
Elena Burceanu*[1,3]
eburceanu@bitdefender.com
Marius Leordeanu[1,2,4]
leordeanu@gmail.com

[1] Bitdefender, Romania
[2] University Politehnica of Bucharest, Romania
[3] University of Bucharest, Romania
[4] Institute of Mathematics of the Romanian Academy, Romania

Abstract

The human ability to synchronize the feedback from all their senses inspired recent works in multi-task and multi-modal learning. While these works rely on expensive supervision, our multi-task graph requires only pseudo-labels from expert models. Every graph node represents a task, and each edge learns between tasks transformations. Once initialized, the graph learns self-supervised, based on a novel consensus shift algorithm that intelligently exploits the agreement between graph pathways to generate new pseudo-labels for the next learning cycle. We demonstrate significant improvement from one unsupervised learning iteration to the next, outperforming related recent methods in extensive multi-task learning experiments on two challenging datasets. Our code is available at https://github.com/bit-ml/cshift.

1 Introduction

Seeing the world from multiple perspectives offers a rich source of knowledge, as recent works show [1, 7, 11, 13, 30, 31]. While multi-tasks methods attain a comprehensive understanding of the scene, they require a larger amount of supervision than single-task ones. Different from multi-modal [19, 20] and multi-task graph approaches [12, 32], we overcome the expensive labeling problem in two steps, taking advantage of existing experts in the literature, pretrained for different tasks. We use them to initially train a multi-task graph, where every node represents a task, and each edge transforms one task into another. We use the mutual consensus along different paths reaching a given task as a self-supervisory signal to further improve over the initial experts. We differ from the most related works [12, 32] in two important ways: 1) we show that without access to labeled data for the target domain, we can initialize the graph using experts from other domains and significantly improve over their performance; 2) our intelligent consensus-finding selection procedure, CShift, which adaptively considers the importance of each incoming edge to a node, is significantly more effective than simple averaging with fixed graph structure [12]. Our **key contributions** are:

1. **We present iterative Consensus Shift (CShift)**, a method for unsupervised multi-task learning for new, unseen data distributions. CShift exploits, with an adaptive edge selection procedure, the consensus among multiple pathways reaching a given node (task), which

* Equal contribution.

	Supervision on target domain	Uses ensembles	Ensemble as supervision	Selection in ensemble	Unsupervised domain adapt.	Full graph
XTC [32]	supervised	✗	N/A	N/A	✗	✓
NGC [12]	semi-supervised	✓	✓	✗	✗	✗
CShift (ours)	unsupervised	✓	✓	✓	✓	✓

Table 1: Learning in Multi-Task Graphs. CShift algorithm vs State-of-the-Art methods.

becomes a supervisory signal at that node. Learning continues over multiple iterations, during which node pseudo-labels shift their values and improve accuracy after each iteration. While initial labels are provided by experts pre-trained on other datasets, the multi-task graph successfully adapts to the new data distribution.

2. **We validate our claims in extensive experiments** on two recent datasets. CShift self-improves in an unsupervised manner, over multiple tasks, from one iteration to the next, while significantly outperforming the state-of-the-art experts used for the initial pseudo-labels.

1.1 Relation to prior work

Relation to Ensembles and Experts. The idea of many paths working together to reach a common goal was often demonstrated [23] over time. We guide learning using a set of expert models, which has proven effective on video and image retrieval [5, 13, 15, 18].

Relation to Unsupervised Representation Learning. Recent works use pretext tasks [6, 17, 34, 35], perform clustering [2, 27, 33], minimize contrastive noise [8, 16, 26] or train adversarial generative models [9]. Different from them, we train on pseudo-ground truth, constructed with our CShift algorithm, from multiple graph paths reaching the same task.

Relation to Multi-Modal Learning. Recent papers combine modalities and tasks [20], often using using multi-modal data transformation as self-supervision [16, 19, 25]. Different from them, we learn complex interactions between tasks without ground truth by modeling them with a bidirectional multi-task graph, using multi-path consensus and selection as supervision.

Relation to Unsupervised Domain Adaptation (UDA). A common UDA approach is to reduce discrepancy in feature space [14, 28]. Others use adversarial training with semantic and style consistency [9, 10]. Ours directly transforms between multiple tasks and exploits across task pixel-level consistencies. The method in [21] learns across two tasks in a particular setup with supervision for both in the source domain and only for one in the target domain. An extension with adversarial training is introduced in [3]. In our case, we do not require annotation for the target domain and rely entirely on consensus between multiple tasks.

Relation to Multi-Task Learning. [31, 32] shows that underlying connections between different tasks can be exploited to effectively reduce the labeled data required for training. We, however, assume no multi-task annotated data on the target domain but only rely on per-task expert models, pre-trained on different domains, for initialization. After initialization, our CShift learns, in fact, completely unsupervised on the target domain. From the architectural point of view, our model is related to the recent Neural Graph Consensus (NGC) model [12], which also connects multiple interpretations and tasks into a single graph of neural networks. Our model differs from NGC in four essential aspects: **1)** the proposed selection mechanism is highly adaptive (different for each pixel, sample, and iteration), allowing a dynamical adjustment of the graph structure as detailed in Sec. 2.2 and Fig. 3, compared with the simple average in NGC; **2)** our fully connected graph guides the learning process only based on unsupervised consensus, as opposed to having a fixed, pruned architecture based on supervised

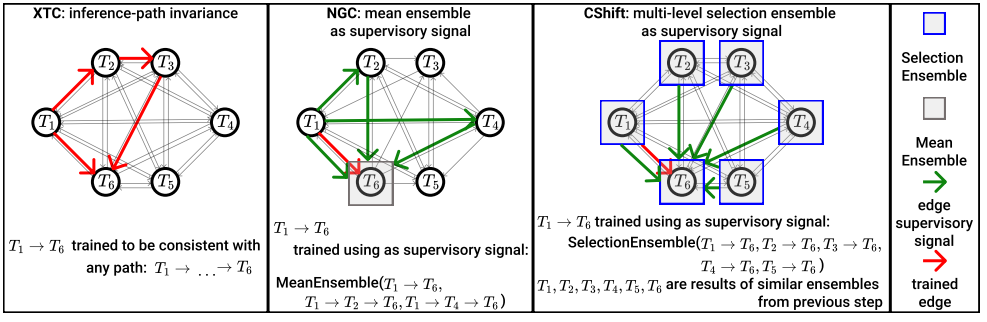

Figure 1: Training strategies employed by other Multi-Task Graph methods compared with CShift. Different from NGC, our selection is unsupervised and uses the consensus of all the graph's edges. Node views are replaced with the consensus of the in-edges (blue box) and ensemble results become both inputs of out-edges and supervisory signals in the next iterations, generating a faster knowledge propagation.

data (NGC); **3)** we use the ensemble labels both as a supervisory signal at a node and as input for all its out-edges, making learning more efficient; **4)** we initialize the graph with pseudo-labels generated by out-of-distribution experts, while NGC assumes a fully supervised initialization. In Tab. 1 and Fig. 1 we show key differences against XTC [32] and NGC [12].

2 Our Approach

We propose a novel Multi-Task Graph (Fig. 2-a), which uses as supervision the consensual output, extracted through an intelligent CShift selection procedure (Fig. 2-c), over multiple graph pathways that reach a given node. As previously mentioned, each graph node represents a task (or a view of the world). Each edge is a neural net that transforms a task at one node into another at a different node. Our graph is directed and fully connected. In Fig. 2 we illustrate the main steps of our approach. All edges are initially trained using node pseudo-labels generated by out-of-distribution experts. After initialization (Fig. 2-b), we set up the **view associated with a node**, computed as the CShift ensemble result of all the node's in-edges. These views become the pseudo-ground truth labels during the subsequent learning iterations in the graph. The views (pseudo-label values) shift from one learning iteration to the next, according to the CShift algorithm. CShift uses the views at each node to transmit information through the out-edges towards other nodes and collects the in-edges' information to create the new views, at the next iteration, by a selection mechanism that establishes the consensus over the multiple incoming edges. The process is repeated until equilibrium is found at convergence. The iterative learning phase is unsupervised and initial experts could be created independently, using information from other datasets and domains, as our tests show.

2.1 Multi-Task Graph

We formally define the Multi-Task Graph over a set T of tasks (*e.g.* semantic segmentation, single-image depth estimation, surface normals - Sec. 3), each illustrating a different view of the scene. There are one-to-one correspondences between graph nodes and the set of tasks, and each edge is an encoder-decoder neural net transformation between source and target task

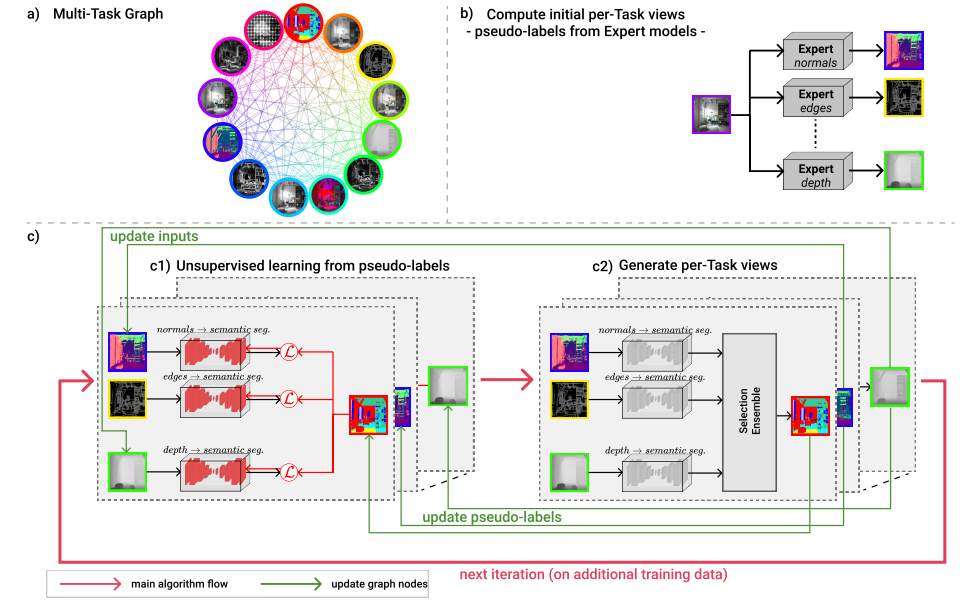

Figure 2: **CShift** architecture. **a) Fully-connected Multi-Task Graph**, with 13 nodes (tasks). Edges are neural nets, transforming source to destination tasks. **b) Initialization from experts.** Based on the *rgb* image, experts (black boxes in our system, trained on different distributions than ours) predict the initial pseudo-labels for each task. **c) CShift's iterations.** We train each graph edge using pseudo-labels. For each node, we compute its new pseudo-labels as the consensual representations of its in-edges by the intelligent CShift ensemble mechanism, which adaptively changes (differently for each pixel and data sample) the importance of each in-edge in the ensemble. The newly computed labels of a node become its supervisory signal in the next iteration and also inputs for all the out-edges of the node.

nodes (Fig. 2-a). Consequently, our graph $G = (T, E)$ with $E = \{e_{s\to d} | e_{s\to d}(X_s) = X_d, s, d \in T, s \neq d\}$, where X_s is the scene representation under task s, and $e_{s\to d}$ is the neural network transforming the view between source task s and destination task d. The graph edges are initialized using pre-trained expert teachers, one for each considered task.

Passing an image through the graph: given a raw *rgb* frame, we associate it to the *rgb* node. Next, we aim to enforce the consensual constraint in the graph: no matter what path the input *rgb* takes through the graph, being transformed from one node to the next, it should have the same representation (view) at the same final node.

Initializing the graph: Other approaches [12, 32] start with a supervised training phase of all the out-edges of the *rgb* node $\{e_{rgb\to d} | \forall d \in T\}$, requiring multi-task annotated datasets. As we work in the unsupervised regime for the target domain, our initial views (pseudo-labels) for different tasks are obtained from a set of out-of-distribution expert models (Fig. 2-b). Each task node d has an associated expert: Expert_d. Then the initial edges are trained by distilling the knowledge of the experts (see the list of experts in Sec. 3).

2.2 Consensus Shift Learning

The consensus between edges reaching a given task d provides a robust view for d. With each learning cycle, the node values (pseudo-labels) shift towards stronger consensus, following the

Algorithm 1 - CShift: Multi-Task Graph Learning with Consensus Shift

X_i - input data sample i | $Y_{i;d}$ - pseudo-label for data i, task d
Expert_d - expert for task d | $e_{s\to d}$ - NN edge from task s to task d
$P = \bigcup_{k=1}^{n_{iters}} part_k$ - dataset split | T - the set of all tasks
$\mathbf{W}_{i;d}$ - per-pixel Selection Ensemble weights, for sample i and destination task d

Results: 1) CShift node views $Y_{i;d}$; 2) all trained edges $e_{s\to d}$

$Y_{i;d} \leftarrow \text{Expert}_d(X_i), \forall d \in T, \forall i \in P$ // Fig. 2-b) Generate the initial pseudo-labels
for $k \leftarrow 1$ to n_{iters} **do**
  $X_{i;s} \leftarrow Y_{i;s}, \forall s \in T, \forall i \in P$ // Update views - enable multi-level ensembles
  for all $d \in T$ **do**
    train $e_{s\to d}(X_{i;s}) = X_{i;d}, \forall i \in part_k, \forall s \in T$ // Fig. 2-c1) Train with pseudo-labels
    $Y_{i;d} \leftarrow f(\bigcup_s \{e_{s\to d}(X_{i;s})\} \cup \{X_{i;d}\}, \mathbf{W}_{i;d}), \forall i \in P$ // Fig. 2-c2) Selection ensemble
  end for
end for

CShift algorithm (Alg. 1). The new labels are then used to distill the single edges connecting them and thus set up the next learning stage. Each transformation is, in fact, the last step of a longer graph path, starting in the *rgb* node and ending in a destination node d. All paths should ideally be in consensual agreement, but in practice, they are not, so an intelligent mechanism is needed to extract the robust knowledge shared by the majority. We employ the discovery of consensus among multiple outputs from incoming edges. Intuitively, CShift is an adaptive combination over the output of all edges reaching a destination. It is based on a similarity measure between those views, computed at pixel-level, which adaptively estimates the importance of each incoming edge, dynamically for each pixel and data sample.

Given a destination node d, all edges reaching this node $\{e_{s\to d} | s \in T, s \neq d\}$ are transformations from different views towards task d. For a sample X_i, CShift iteratively updates the sample's view, associated with task d, $X_{i;d}$. We define $\mathcal{N}(X_{i;d})$ to be the neighbourhood of $X_{i;d}$ as the set of all transformations from all different views of the sample joined with the current pseudo-label: $\mathcal{N}(X_{i;d}) = \{e_{s\to d}(X_{i;s}) | \forall s \in T\} \cup \{X_{i;d}\}$. The current task representation is replaced by the consensual one, computed as a function f gathering information from all the neighbours of $X_{i;d}$, parameterized by pixel-level weights $\mathbf{W}_{i;d} \in \mathbb{R}^{h\times w\times |T|}$ ((h,w) is the image size): $X_{i;d} \leftarrow f(\mathcal{N}(X_{i;d}); \mathbf{W}_{i;d})$, capturing the consensus between predictions. $\mathbf{W}_{i;d}$ has a channel associated with each task node, which indicates the similarity of the corresponding prediction with the current value of d. Without loss of generality, we assume d is a single channel task. For a location (x,y) and a given task s, the weights are computed as follows:

$$\mathbf{W}_{i;d}[x,y,s] = \frac{K(dist(e_{s\to d}(X_{i;s}), X_{i;d})[x,y])}{\sum_{Z\in\mathcal{N}(X_{i;d})} K(dist(Z, X_{i;d})[x,y])}, \tag{1}$$

where $dist : \mathbb{R}^{h\times w} \to \mathbb{R}^{h\times w}$ is a distance function capturing the similarity between two different prediction maps and $K : \mathbb{R} \to \mathbb{R}$ is the kernel function that determines the weight of nearby points. The algorithm aims to identify the areas of the prediction maps that are perceptually similar and push them further in the ensemble while downgrading regions that seem to be noisy and uncorrelated with the other predictions. We propose a selection ensemble algorithm that automatically extracts the most representative consensual representation of the in-edges, being adaptive and changing separately for each pixel and data sample - essentially changing the graph's structure in a dynamic and per-pixel manner. The adaptive per-pixel weighting

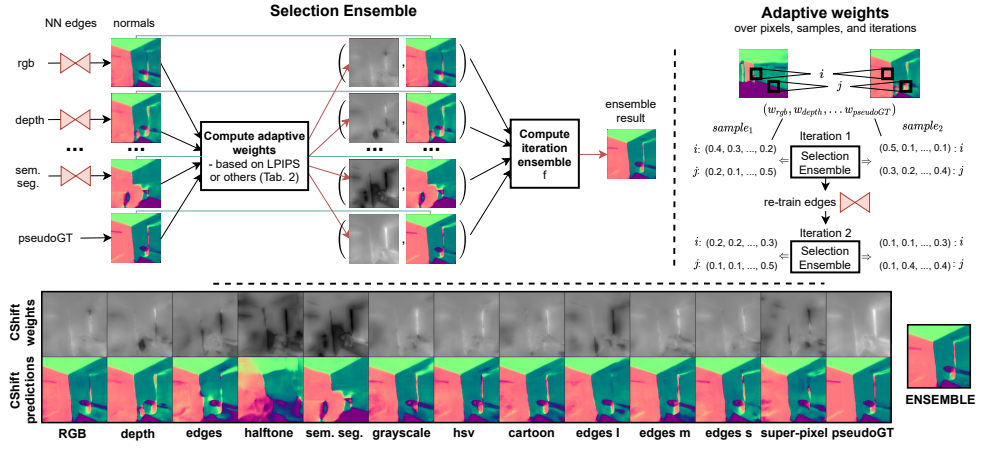

Figure 3: CShift selection flow, stripped down to pixel-level. The complex weights are highly adaptive, modifying, in effect, the underneath graph structure through their values. The last two rows show CShift per-pixel weights and the corresponding predictions from source tasks to *normals* task destination. Note how the sharp zones correlate with high ensemble weights.

allows CShift to keep the most relevant information from all input maps, even when some maps are less reliable, treating similarities per region (kernels at pixel-level). In Sec. 3 we instantiate $dist, K, f$ and provide ablation experiments proving that the selection strategy is robust to noisy connections. We show in Fig. 3 how the selection based consensus works at pixel-level.

3 Experimental Analysis

Datasets. We perform experiments on Replica [24] and Hypersim [22]. Replica is a dataset of photo-realistic 3D indoor scenes, comprising 18 scenes, with a total of 48 rooms. In practice, we consider two iterations for training our Multi-Task Graph and use two unsupervised train sets (9600+9600 samples), a validation and a test set (each with 960 samples). Hypersim is also a photorealistic synthetic dataset, for holistic indoor scene understanding. For two iteration training we use 10696 + 10818 samples, 319 for validation and 1064 for testing. During training and validation, only the raw *rgb* images are available. We highlight that the annotations are only employed for evaluation purposes.

Expert Models. We employ per-task expert models, having only *rgb* as input, trained on out-of-distribution data, to initialize the per-node pseudo-labels (Fig. 2-b). First, the experts replace the direct edges starting from the *rgb* node. Then, the computed views of a node become supervisory signals for the in-edges of that node and inputs for the out-edges (*e.g.* for training edge $e_{depth \to halftone}$, the depth input is obtained by applying the depth expert over the *rgb* frame and the halftone pseudo-label by extracting the halftone from *rgb*.) Our graph contains a total of 13 task nodes, including *rgb*, thus we consider 12 experts ranging from trivial color-space transformations to heavily trained deep nets for the following tasks: **1)** halftone, **2)** grayscale, **3)** hsv, **4)** depth, **5)** surface normals, **6, 7, 8)** small, medium and large scale low-level edges, **9)** high-level edges, **10)** super-pixel, **11)** cartoonization and **12)** semantic segmentation. The experts are trained on a wide variety of datasets, having a different distribution than ours. We detail the expert models in the supplementary material.

Method	*depth*	*normals*	*rgb*
Expert [32]	14.58	8.30	-
Mean Ensemble [12]	12.94	7.95	4.30
CShift w/ Variance	12.80	7.91	2.12
CShift w/ PSNR	12.89	8.12	4.25
CShift w/ SSIM	12.80	7.89	2.38
CShift w/ L1	12.81	7.73	2.16
CShift w/ L2	12.79	7.72	2.45
CShift w/ LPIPS	**12.77**	**7.61**	**2.06**

Table 2: Ablation study on different distance metrics on Replica dataset, for the first iteration. In all considered configurations CShift overcomes the initial expert models and in all, except the PSNR case, the Mean Ensemble.

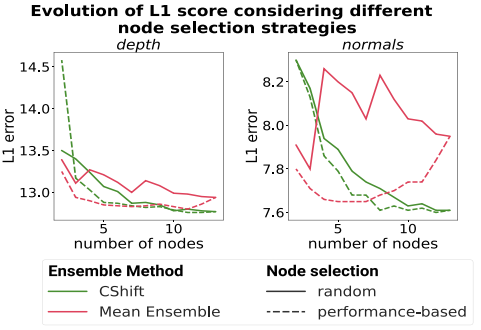

Figure 4: CShift is stable under different node selection strategies, improving even when using low-performing edges. In contrast, the mean ensemble has an unstable evolution under random node selection, its performance decreasing when weak edges are added.

Evaluation on three tasks. Replica and Hypersim have similar annotation conventions only for *depth* and *normals* tasks. The XTC experts' considered for the tasks are not fully aligned with testing datasets annotations, as their training dataset uses different conventions. Thus, on *depth*, following the methodology of self-supervised methods [37], we performed a histogram specification alignment between expert results and ground truth annotations. On *normals*, we removed the 3rd channel in the XTC expert as Replica has *normals* with only 2 independent channels. We also report results for *rgb*, measuring the graph model's ability to reconstruct its original input through its many paths. We report the L1 error $\times 100$ (for readability).

Implementation and training details. Each graph edge is a neural network with a UNet architecture, as previously validated in NGC and XTC. The 156 graph edges have ≈ 4.3 million parameters each, with 4 down-scaling and 4 up-scaling layers and a proper number of input and output channels, depending on the source and destination tasks. We optimize them by jointly minimizing *L2* and the Structural Similarity Index Measure [29] (SSIM) losses for regression tasks equally weighted for each neural net. For training edges going to classification tasks (semantic segmentation or halftone), we use Cross-Entropy loss. As optimizer we work with SGD with Nesterov (lr=5e-2, wd=1e-3, momentum=0.9), and a ReduceLROnPlateau scheduler (patience=10, factor=0.5, threshold=1e-2, min lr=5e-5). We train 100 epochs for the 1st iteration and 100 for the 2nd one. Note that for the second iteration, we train all edges from scratch.

Ensemble Selection Method. At the core of the selection procedure is the distance function dictating each prediction map's per-pixel weights, as detailed in Sec. 2.2. We instantiate f to the weighted median, and the kernel function K (Eq. 1) to identity. To understand the selection strategy's power, we consider distance metrics ranging from local per-pixel distances to global perceptual measures: **1)** L1 and **2)** L2 distances at pixel-level **3)** Peak signal-to-noise ratio (PSNR) to measure the noise of the predictions; **4)** Structure Similarity Index Measure (SSIM) [29] that analyses the luminance, contrast and structural differences; **5)** Learned Perceptual Image Patch Similarity (LPIPS) [36] that is a deep model trained to identify perceptually similar images; **6)** per-pixel Variance among the multi-path predictions, to quantify their consensus. In Tab. 2 we compare 1st CShift iteration under different selection strategies. Our proposed selections overcome the expert models and the simple mean ensemble

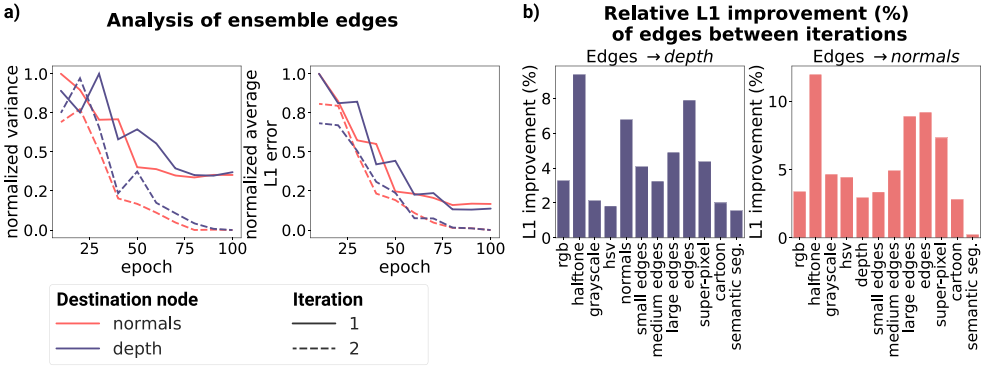

Figure 5: **a)** The left plot shows that the average variance over single edges in an ensemble decreases over the training epochs and iterations. Thus, the graph's average consensus improves from one iteration to the next, while the average L1 error of edges decreases (right). CShift is effective and edges evolve towards the GT rather than collapsing in trivial solutions. **b)** We present the relative performance improvement of the individual edges between iterations. The performance of each edge increases up to almost 12%, proving the capacity of CShift to iteratively adapt to the new domain, in an unsupervised manner.

in almost all the considered configurations (except for PSNR), highlighting the robustness of the process. Note that the simple mean ensemble baseline is similar to some extent with NGC. Our models are only trained using the expert models while NGC employs a supervised initialization step. We chose for all the subsequent experiments CShift w/ LPIPS.

Consensus under different sets of nodes. To validate that our model is robust to the set of the considered nodes (and their corresponding in and out edges), we perform an experiment where we start with a small graph containing only two nodes, and, step-by-step, increase the number of nodes until reaching all 13 nodes. The nodes are added to the graph in a specific order. We analyze two ways of establishing this order: 1) random - nodes are randomly sorted; 2) performance-based - nodes are sorted according to their individual performance (evaluated w.r.t. to ground truth annotations) and added in this order. In Fig. 4 we present the results of our experiment, comparing CShift with a mean ensemble baseline, for two destination tasks: *depth* and *normals*. In both scenarios, the performance of CShift increases with the number of nodes, proving that our ensemble selection mechanism is able to extract relevant information even from low-performing edges. Note the performance fluctuations in baseline showing it is highly dependent on each edge reaching the ensemble.

Edges improvement between iterations. We give an in-depth analysis of how individual edges evolve over training epochs and CShift iterations. First, in Fig. 5-a), we see how the variance between the edges in an ensemble decreases with more training. This is natural since each edge in the ensemble uses the same pseudo-GT. But, in the second iteration, the variance is even smaller, showing a smoother training optimization for the edges (which are trained from scratch for this second iteration). This could be explained by the new pseudo-labels coming from iteration 1 ensembles, rather than experts, making the training process simpler. To validate that the edges do not collapse to a bad representation, we also plot the average L1 error in Fig. 5-a), confirming that all the edges improve their performance towards the GT. We show those relative improvements per edge, between the two CShift iterations in Fig. 5-b).

Qualitative views for multiple tasks. In Fig. 6-a), we show the differences between the expert output, used as initial pseudo-ground truth for our graph edges (Fig. 2-b), and the

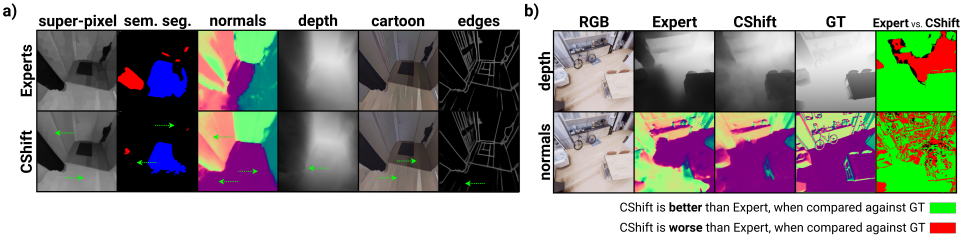

Figure 6: **a)** We show the outputs of experts' used as pseudo-labels (1st row) and of CShift (2nd row). Green arrows point to significant improvements. From left to right are the representation for multiple tasks: super-pixel and cartoonization have less noise; semantic segmentation removes almost all pixels wrongly classified as ceiling; surface normals are significantly corrected; depth catches new details from the curtain; on edges, it removes some of the noisy edges coming from the floor texture. **b)** We compare the Expert with CShift (col 2-3). In the last column, we show the differences, green for pixels with improved prediction and red for weaker ones. CShift (second row) adds a bike in the scene, where the Expert completely misses it. The tasks are naturally interconnected and CShift takes advantage of that.

	Method	**Replica** $dest_{task}$ (L1 ↓)			**Hypersim** $dest_{task}$ (L1 ↓)		
		depth	*normals*	*rgb*	*depth*	*normals*	*rgb*
	Expert [32]	14.58	8.30	-	15.11	**9.10**	-
Iter 1	Average of direct edges	14.32	9.34	6.33	16.91	12.55	11.34
	Edge: $rgb \rightarrow dest_{task}$	13.42	8.23	-	15.97	11.75	-
	Mean Ensemble [12]	12.94	7.95	4.30	14.84	10.56	8.22
	CShift	12.77	**7.61**	2.06	13.98	9.36	3.56
Iter 2	Average of direct edges	13.70	8.83	5.00	15.74	11.37	9.01
	Edge: $rgb \rightarrow dest_{task}$	**12.98**	**7.95**	-	**15.03**	10.09	-
	Mean Ensemble	12.87	7.91	3.18	14.20	9.90	6.31
	CShift	**12.71**	**7.61**	**1.51**	**13.75**	**9.02**	**1.84**
	CShift Boost ↑	**12.8%**	**8.3%**	-	**9.0%**	**0.9%**	-

Table 3: Quantitative results. We compare our performance over each iteration against the initial experts, on destination tasks for which we have GT annotations. CShift ensemble outperforms XTC experts on *depth* and *normals* by a large margin, without any additional supervision. Even single, direct edges ($rgb \rightarrow dest_{task}$) improve over iterations, achieving better results compared with the experts in most of the cases (except for Hypersim's *normals*). With blue we represent the best single edge in the column and with red the best ensemble.

output of our CShift algorithm. Notice that the output of CShift looks smoother and partially corrects the mistakes of the expert model, adding significant value to the output.

Qualitative results for tasks with GT. We compare next in Fig. 6-b) our results with the Expert, w.r.t. ground truth. Notice that CShift improves the output at a profound level, bringing in new information in the scene (see the bike in the second row). This is due to the multiple different source tasks for the ensembles' in-edges.

Comparison with other methods. Starting from pseudo-labels provided by the experts, we

$dest_{task}$: *sem. seg.*	**Expert models** (L1 ↓) (from 1-weak to 6-strong)					
Method	1	2	3	4	5	6
Expert	2.9	2.5	2.2	1.8	1.6	1.5
Mean Ensemble	2.6	2.5	2.3	2.1	1.9	2.0
CShift	2.4	2.3	2.0	1.7	1.5	1.4
CShift Boost ↑	15.7%	10.6%	8.2%	6.3%	6.1%	5.5%

Table 4: The boost over the expert varies inversely with the expert strength (from left to right). As expected, it is harder to improve over a good expert, but CShift does it in all tests.

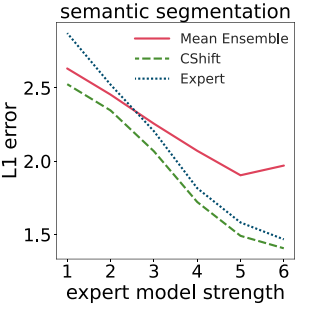

Figure 7: CShift consistently improves over experts of different quality.

improve their quality by a large margin with our CShift ensemble and even with a direct link from *rgb* (except for Hypersim's normals where GT is extremely detailed, while our single edge is a simple UNet with 4.3 mil params). We achieve this performance (Tab. 3) in just two CShift iterations, without adding any supervised information, largely outperforming the basic mean ensemble. Notice that the direct edges in the graph significantly improves over CShift iterations (in average and the individual direct edge from *rgb*). We treat all tasks unitary, so we tested the *rgb* reconstruction performance, noticing large improvements over iterations.

Using weak expert models. We test the importance of using weaker experts and expand the quantitatively validated domains set on semantic segmentation tasks (Hypersim dataset, with 40 classes). To control the level of expertise of our initial expert, we have trained it from scratch in a supervised manner on randomly chosen 10%, 30% and 50% of training samples of Hypersim, for 30 and 40 epochs, resulting in 6 expert models, with increasing performance. The relatively small training sets and the number of training epochs are specifically chosen to ensure that the experts are weak classifiers. This case is different and complementary to the other experiments in which we brought state-of-the-art experts pretrained on other datasets. Then we use these weak "expert" models, one at a time, as the regular experts in our first CShift iteration. We report L1 errors of class probability maps consistent with all the other tasks. In all cases, CShift outperforms both the initial expert and the baseline mean. A remarkable observation is that the gap between the expert and CShift varies inversely with the power of the expert: the weaker the expert, the higher the improvement (Tab. 4, Fig. 7).

4 Concluding remarks

We introduce the CShift algorithm in multi-task graphs, able to learn unsupervised in new data distributions, using as supervision an intelligent consensus among the multiple pathways reaching a given task node, which adapts for each individual pixel and data sample. CShift's unsupervised capability and intelligent per-pixel ensemble selection for creating pseudo-labels make it significantly different and stronger than related methods. All key aspects of our approach, namely the CShift selection ensemble, the unsupervised domain adaptation capability, and the ability to learn from weak experts, are experimentally validated on two challenging datasets. Also, the comparisons to recent related works prove superior capabilities in the unsupervised learning case. We believe that CShift brings theoretically interesting and practically valuable contributions in an area of research, that of multi-task unsupervised learning, which is of major importance in today's machine learning.

Acknowledgements

This work was funded in part by UEFISCDI, under Projects EEA-RO-2018-0496 and PN-III-P4-ID-PCE-2020-2819.

References

[1] William H. Beluch, Tim Genewein, Andreas Nürnberger, and Jan M. Köhler. The power of ensembles for active learning in image classification. In *CVPR*, 2018.

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arxiv*, 2020.

[3] Ruchika Chavhan, Ankit Jha, Biplab Banerjee, and Subhasis Chaudhuri. Ada-at/dt: An adversarial approach for cross-domain and cross-task knowledge transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3502–3511, 2021.

[4] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *NeurIPS*, 2019.

[5] Matthijs Douze, Arnau Ramisa, and Cordelia Schmid. Combining attributes and fisher vectors for efficient image retrieval. In *CVPR*, 2011.

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

[7] Joris Guérin and Byron Boots. Improving image clustering with multiple pretrained CNN feature extractors. In *BMVC*, 2018.

[8] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010.

[9] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.

[10] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

[11] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018.

[12] Marius Leordeanu, Mihai Pirvu, Dragos Costea, Alina Marcu, Emil Slusanschi, and Rahul Sukthankar. Semi-supervised learning for multi-task scene understanding by neural graph consensus. In *AAAI*, 2021.

[13] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *BMVC*, 2019.

[14] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[15] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arxiv*, 2018.

[16] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020.

[17] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *ECCV*, 2016.

[18] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K. Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *ICMR*, 2018.

[19] Mandela Patrick, Yuki Markus Asano, Ruth Fong, João F. Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. *arxiv*, 2020.

[20] A. J. Piergiovanni, Anelia Angelova, and Michael S. Ryoo. Evolving losses for unsupervised video representation learning. In *CVPR*, 2020.

[21] Pierluigi Zama Ramirez, Alessio Tonioni, Samuele Salti, and Luigi Di Stefano. Learning across tasks and domains. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8110–8119, 2019.

[22] Mike Roberts and Nathan Paczan. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. arXiv, 2020.

[23] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Data Min. Knowl. Discov.*, 2018.

[24] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv*, 2019.

[25] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arxiv*, 2019.

[26] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *CoRR*, 2020.

[27] Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. Unsupervised learning of video representations via dense trajectory clustering. *CoRR*, 2020.

[28] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[29] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[30] Meng Ye and Yuhong Guo. Progressive ensemble networks for zero-shot recognition. In *CVPR*, 2019.

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

[32] Amir Roshan Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J. Guibas. Robust learning through cross-task consistency. In *CVPR*, 2020.

[33] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *CVPR*, 2020.

[34] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *ECCV*, 2016.

[35] Richard Zhang, Phillip Isola, and Alexei A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.

[36] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[37] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017.