Robust Learning Through Cross-Task Consistency

Amir R. Zamir†∗ Alexander Sax‡∗ Nikhil Cheerla§ Rohan Suri§ Zhangjie Cao§ Jitendra Malik‡ Leonidas Guibas§

† Swiss Federal Institute of Technology (EPFL)  § Stanford University  ‡ UC Berkeley

http://consistency.epfl.ch/

Abstract

Visual perception entails solving a wide set of tasks, e.g., object detection, depth estimation, etc. The predictions made for multiple tasks from the same image are not independent, and therefore, are expected to be ‘consistent’. We propose a broadly applicable and fully computational method for augmenting learning with Cross-Task Consistency.† The proposed formulation is based on inference-path invariance over a graph of arbitrary tasks. We observe that learning with cross-task consistency leads to more accurate predictions and better generalization to out-of-distribution inputs. This framework also leads to an informative unsupervised quantity, called Consistency Energy, based on measuring the intrinsic consistency of the system. Consistency Energy correlates well with the supervised error ($r=0.67$), thus it can be employed as an unsupervised confidence metric as well as for detection of out-of-distribution inputs (ROC-AUC=$0.95$). The evaluations are performed on multiple datasets, including Taskonomy, Replica, CocoDoom, and ApolloScape, and they benchmark cross-task consistency versus various baselines including conventional multi-task learning, cycle consistency, and analytical consistency.

1. Introduction

What is consistency: suppose an object detector detects a ball in a particular region of an image, while a depth estimator returns a flat surface for the same region. This presents an issue – at least one of them has to be wrong, because they are inconsistent. More concretely, the first prediction domain (objects) and the second prediction domain (depth) are not independent and consequently enforce some constraints on each other, often referred to as consistency constraints.

Why is it important to incorporate consistency in learning: first, desired learning tasks are usually predictions of different aspects of one underlying reality (the scene that underlies an image). Hence inconsistency among predictions implies contradiction and is inherently undesirable. Second, consistency constraints are informative and can be used to better fit the data or lower the sample complexity. Also, they may reduce the tendency of neural networks to learn “surface statistics” (superficial cues) [18], by enforcing constraints rooted in different physical or geometric rules. This is empirically supported by the improved generalization of models when trained with consistency constraints (Sec. 5).

How can we design a learning system that makes consistent predictions: this paper proposes a method which, given an arbitrary dictionary of tasks, augments the learning objective with explicit constraints for cross-task consistency. The constraints are learned from data rather than apriori given relationships. This makes the method applicable to any pairs of tasks as long as they are not statistically independent; even if their analytical relationship is unknown, hard to program, or non-differentiable. The primary concept behind the method is ‘inference-path invariance’. That

1Abbreviated X-TC, standing for Cross-Task Consistency.

*Equal.

2For instance, it is not necessary to encode that surface normals are the 3D derivative of depth or occlusion edges are discontinuities in depth.
is, the result of inferring an output domain from an input domain should be the same, regardless of the intermediate domains mediating the inference (e.g., RGB→normals and RGB→depth→normals and RGB→shading→normals are expected to yield the same normal results). When inference paths with the same endpoints, but different intermediate domains, yield similar results, this implies the intermediate domain predictions did not conflict as far as the output was concerned. We apply this concept over paths in a graph of tasks, where the nodes and edges are prediction domains and neural network mappings between them, respectively (Fig. 2(d)). Satisfying this invariance constraint over all paths in the graph ensures the predictions for all domains are in global cross-task agreement.3

To make the associated large optimization job manageable, we reduce the problem to a ‘separable’ one, devise a tractable training schedule, and use a ‘perceptual loss’ based formulation. The last enables mitigating residual errors in networks and potential ill-posed/one-to-many mappings between domains (Sec. 3).

Interactive visualizations, trained models, code, and a live demo are available at http://consistency.epfl.ch/.

2. Related Work

The concept of consistency and methods for enforcing it are related to various topics, including structured prediction, graphical models [22], functional maps [30], and certain topics in vector calculus and differential topology [10]. We review the most relevant ones in context of computer vision.

Utilizing consistency: Various consistency constraints have been commonly found beneficial across different fields, e.g., in language as ‘back-translation’ [2, 1, 25, 7] or in vision over the temporal domain [41, 6], 3D geometry [9, 32, 13, 49, 46, 15, 44, 51, 48, 23, 5], and in recognition and (conditional/unconditional) image translation [12, 28, 17, 50, 14, 4]. In computer vision, consistency has been extensively utilized in the cycle form and often between two or few domains [50, 14]. In contrast, we consider consistency in the more general form of arbitrary paths with varied-lengths over a large task set, rather than the special cases of short cyclic paths. Also, the proposed approach needs no prior explicit knowledge about task relationships [32, 23, 44, 51].

Multi-task learning: In the most conventional form, multi-task learning predicts multiple output domains out of a shared encoder/representation for an input. It has been speculated that the predictions of a multi-task network may be automatically cross-task consistent as the representation from which the predictions are made are shared. This has been observed to not be necessarily true in several works [21, 47, 43, 38], as consistency is not directly enforced during training. We also make the same observation (see visuals here) and quantify it (see Fig. 9(a)), which signifies the need for explicit augmentation of consistency in learning.

Transfer learning predicts the output of a target task given another task’s solution as a source. The predictions made using transfer learning are sometimes assumed to be cross-task consistent, which is often found to not be the case [45, 36], as transfer learning does not have a specific mechanism to impose consistency by default. Unlike basic multi-task learning and transfer learning, the proposed method includes explicit mechanisms for learning with general data-driven consistency constraints.

Uncertainty metrics: Among the existing approaches to measuring prediction uncertainty, the proposed Consistency Energy (Sec. 4) is most related to Ensemble Averaging [24], with the key difference that the estimations in our ensemble are from different cues/paths, rather than retraining/reevaluating the same network with different random initializations or parameters. Using multiple cues is expected to make the ensemble more effective at capturing uncertainty.

3. Method

We define the problem as follows: suppose \( x \) denotes the query domain (e.g., RGB images) and \( y = \{y_1, ..., y_n\} \) is the set of \( n \) desired prediction domains (e.g. normals, depth, objects, etc). An individual datapoint from domains \( (x, y_1, ..., y_n) \) is denoted by \( (x, y_1, ..., y_n) \). The goal is to learn functions that map the query domain onto the prediction domains, i.e., \( \mathcal{F}_x = \{f_{xy_j} | y_j \in y\} \) where \( f_{xy_j}(x) \) outputs \( y_j \) given \( x \). We also define \( \mathcal{F}_y = \{f_{y_j}, y_j \in y, t \neq j\} \), which is the set of ‘cross-task’ functions that map the prediction domains onto each other; we use them in the consistency constraints. For now assume \( \mathcal{F}_y \) is given apriori and frozen; in Sec. 3.3 we discuss all functions \( f \) are neural networks in this paper, and we learn \( \mathcal{F}_y \) just like \( \mathcal{F}_x \).

3.1. Triangle: The Elementary Consistency Unit

The typical supervised way of training the neural networks in \( \mathcal{F}_x \), e.g. \( f_{xy_1}(x) \), is to find parameters of \( f_{xy_1} \) that minimize a loss with the general form \( \|f_{xy_1}(x) - y_1\| \) using a
distance function as $|.|$, e.g. $\ell_1$ norm. This standard independent learning of $f_{XY}$ satisfies various desirable properties, including cross-task consistency, if given infinite amount of data, but not under the practical finite data regime. This is shown in Fig. 3 (upper). Thus we introduce additional constraints to guide the training toward cross-task consistency. We define the loss for predicting domain $Y_1$ from $X$ while enforcing consistency with domain $Y_2$ as a directed triangle depicted in Fig. 2(b):

$$\mathcal{L}_{X \rightarrow Y_2}^{\triangle} = |f_{XY_1}(x) - y_1| + |f_{XY_2}(x) - f_{XY_1}(x)| + |f_{XY_2}(x) - y_2|. \quad (1)$$

The first and last terms are the standard direct losses for training $f_{XY_1}$ and $f_{XY_2}$. The middle term is the consistency term which enforces predicting $Y_2$ out of the predicted $Y_1$ yields the same result as directly predicting $Y_2$ out of $X$. Thus learning to predict $Y_1$ and $Y_2$ are not independent anymore.

The triangle loss 1 is the smallest unit of enforcing cross-task consistency. Below we make two improving modifications on it via function ‘separability’ and ‘perceptual losses’.

### 3.1.1 Separability of Optimization Parameters

The loss $\mathcal{L}_{X \rightarrow Y_1}^{\triangle}$ involves simultaneous training of two networks $f_{XY_1}$ and $f_{XY_2}$, thus it is resource demanding. We show $\mathcal{L}_{X \rightarrow Y_1}^{\triangle}$ can be reduced to a ‘separable’ function [39] resulting in two terms that can be optimized independently.

From triangle inequality we can derive:

$$|f_{XY_2}(x) - f_{XY_1}(x)| \leq |f_{XY_2}(x) - y_2| + |f_{XY_1}(x) - y_2|,$$

which after substitution in Eq. 1 yields:

$$\mathcal{L}_{X \rightarrow Y_1}^{\triangle} \leq |f_{XY_1}(x) - y_1| + |f_{XY_2}(x) - f_{XY_1}(x)| + |f_{XY_2}(x) - y_2|. \quad (2)$$

The upper bound for $\mathcal{L}_{X \rightarrow Y_1}^{\triangle}$ in inequality 2 can be optimized in lieu of $\mathcal{L}_{X \rightarrow Y_1}^{\triangle}$ itself, as they both have the same minimizer.\(^5\) The terms of this bound include either $f_{XY_1}$ or $f_{XY_2}$, but not both, hence we now have a loss separable into functions of $f_{XY_1}$ or $f_{XY_2}$, and they can be optimized independently. The part pertinent to the network $f_{XY_1}$ is:

$$\mathcal{L}_{XY_1 \rightarrow Y_2}^{\text{separate}} \triangleq |f_{XY_1}(x) - y_1| + |f_{XY_2}(x) - f_{XY_1}(x)|, \quad (3)$$

named separate, as we reduced the closed triangle objective $\mathcal{L}_{X \rightarrow Y_1 \rightarrow Y_2}$ in Eq. 1 to two equivalent separate path objectives $X \rightarrow Y_1 \rightarrow Y_2$ and $X \rightarrow Y_2$. The first term of Eq. 3 enforces the general correctness of predicting $Y_1$, and the second term enforces its consistency with $Y_2$ domain.

### 3.1.2 Reconfiguration into a “Perceptual Loss”

Training $f_{XY_1}$ using the loss $\mathcal{L}_{XY_1 \rightarrow Y_2}^{\text{separate}}$ requires a training dataset with multi domain annotations for one input: $(x, y_1, y_2)$. It also relies on availability of a perfect function $f_{Y_2|Y_1}$ for mapping $Y_1$ onto $Y_2$; i.e. it demands $y_2 = f_{Y_2|Y_1}(y_1)$. We show how these two requirements can be reduced.

Again, from triangle inequality we can derive:

$$|f_{XY_2}(x) - y_2| \leq |f_{XY_2}(x) - f_{XY_1}(y_1)| + |f_{XY_2}(y_1) - y_2|, \quad (4)$$

which after substitution in Eq. 3 yields:

$$\mathcal{L}_{XY_1 \rightarrow Y_2}^{\text{separate}} \leq |f_{XY_1}(x) - y_1| + |f_{XY_2}(x) - f_{XY_1}(y_1)| + |f_{XY_2}(y_1) - y_2|. \quad (5)$$

Similar to the discussion for inequality 2, the upper bound inequality 5 can be optimized in lieu of $\mathcal{L}_{XY_1 \rightarrow Y_2}^{\text{separate}}$ as both have the same minimizer.\(^5\) As the last term is a constant w.r.t.

\(^5\)Both sides of inequality 2 are $\geq 0$ and $= 0$ for the minimizer $f_{XY_1}(x) = y_1$ & $f_{XY_2}(x) = y_2$.

\(^5\)Both sides of inequality 5 are $\geq 0$ and $= 0$ for the minimizer $f_{XY_1}(x) = y_1$. The term $|f_{XY_2}(y_1) - y_2|$ is a constant and $\sim 0$, as it is exactly the training objective of $f_{XY_2}$. The non-zero residual should be ignored and assumed 0 as the non-zero part is irrelevant to $f_{XY_1}$, but imperfections of $f_{XY_2}$.

---

**Figure 3:** Impact of disregarding cross-task consistency in learning, illustrated using surface normals domain. Each subfigure shows the results of predicting surface normals out of the prediction of an intermediate domain; using the notation $x \rightarrow y_1 \rightarrow y_2$, here $x$ is RGB image, $y_2$ is surface normals, and each column represents a different $y_1$. The upper row demonstrates the normals are noisy and dissimilar when cross-task consistency is not incorporated in learning of $x \rightarrow y_1$ networks. Whereas enforcing consistency when learning $x \rightarrow y_1$ results in more consistent and better normals (the lower row). We will show this causes the predictions for the intermediate domains themselves to be more accurate and consistent. More examples available in supplementary material. The Consistency Energy (Sec. 4) captures the variance among predictions in each row.
\( f_{XY_1} \), the final loss for training \( f_{XY_1} \) is:

\[
\mathcal{L}_{\text{perceptual}}^{\text{XY_1}}(x; y_1) = \| f_{XY_1}(x) - y_1 \| + \| f_{XY_2} \circ f_{XY_1}(x) - f_{XY_2}(y_1) \|. \tag{6}
\]

The loss \( \mathcal{L}_{\text{perceptual}}^{\text{XY_1}} \) no longer includes \( y_2 \), hence it admits pair training data \((x, y_1)\) rather than triplet \((x, y_1, y_2)\). Comparing \( \mathcal{L}_{\text{perceptual}}^{\text{XY_1}} \) and \( \mathcal{L}_{\text{perceptual}}^{\text{XY_2}} \) shows the modification boiled down to replacing \( y_2 \) with \( f_{Y_2}(y_1) \). This makes intuitive sense too, as \( y_2 \) is the match of \( y_1 \) in the \( Y_2 \) domain.

**Ill-posed tasks and imperfect networks:** If \( f_{XY_2} \) is a noisy estimator, then \( f_{Y_2}(y_1) = y_2 + \text{noise} \) rather than \( f_{Y_2}(y_1) = y_2 \). Using a noisy \( f_{XY_2} \) in \( \mathcal{L}_{\text{perceptual}}^{\text{XY_2}} \) corrupts the training of \( f_{XY_1} \), since the second loss term does not reach 0 if \( f_{XY_1}(x) \) correctly outputs \( y_1 \). That is in contrast to \( \mathcal{L}_{\text{perceptual}}^{\text{XY_1}} \) where both terms have the same global minimum and are always 0 if \( f_{XY_1}(x) \) outputs \( y_1 \) — even when \( f_{Y_2}(y_1) = y_2 + \text{noise} \). This is crucial since neural networks are almost never perfect estimators, e.g. due to lacking an optimal training process for them or potential ill-posedness of the task \( y_1 \rightarrow y_2 \). Further discussion and experiments are available in supplementary material.

**Perceptual Loss:** The process that led to Eq. 6 can be generally seen as using the loss \( \| g \circ f(x) - g(y) \| \) instead of \( \| f(x) - y \| \). The latter compares \( f(x) \) and \( y \) in their explicit space, while the former compares them via the lens of function \( g \). This is often referred to as “perceptual loss” in super-resolution and style transfer literature [19]—where two images are compared in the representation space of a network pretrained on ImageNet, rather than in pixel space. Similarly, the consistency constraint between the domains \( Y_1 \) and \( Y_2 \) in Eq. 6 (second term) can be viewed as judging the prediction \( f_{XY_1}(x) \) against \( y_1 \) via the lens of the network \( f_{XY_2} \):

**3.2. Consistency of \( f_{XY_1} \) with ‘Multiple’ Domains**

The derived \( \mathcal{L}_{\text{perceptual}}^{\text{XY_2}} \) loss augments learning of \( f_{XY_2} \) with a consistency constraint against one domain \( Y_2 \). Straightforward extension of the same derivation to enforcing consistency of \( f_{XY_1} \) against multiple other domains (i.e. when \( f_{XY_1} \) is part of multiple simultaneous triangles) yields:

\[
\mathcal{L}_{\text{perceptual}}^{\text{XY_1}}(x; \{y_1 \}) = \| Y \| \| f_{XY_1}(x) - y_1 \| + \sum_{y_1 \in Y} \| f_{XY_2} \circ f_{XY_1}(x) - f_{XY_2}(y_1) \|, \tag{7}
\]

where \( Y \) is the set of domains with which \( f_{XY_1} \) must be consistent, and \( |Y| \) is the cardinality of \( Y \). Notice that \( \mathcal{L}_{\text{perceptual}}^{\text{XY_1}} \) is a special case of \( \mathcal{L}_{\text{perceptual}}^{\text{XY_2}} \) where \( Y = \{y_2 \} \). Fig. 5 summarizes the derivation of losses for \( f_{XY_1} \).

Fig. 4 shows qualitative results of learning \( f_{XY_1} \) with and without cross-task consistency for a sample query.

**3.3. Beyond Triangles: Globally Consistent Graphs**

The discussion so far provided the loss for the cross-task consistent training of one function \( f_{XY_1} \), using elementary

---

**Figure 4:** Learning with and without cross-task consistency shown for a sample query. Using the notation \( x \rightarrow Y_1 \rightarrow Y \), here \( x \) is RGB image, \( Y_1 \) is surface normals, and five domains in \( Y \) are reshading, 3D curvature, texture edges (Sobel filter), depth, and occlusion edges. Top row shows the results of standard training of \( x \rightarrow Y_1 \). After convergence of training, the predicted normals \( Y_1 \) are projected onto other domains \( Y \) which reveal various inaccuracies. This demonstrates such cross-task projections \( Y_1 \rightarrow Y \) can provide additional cues to training \( x \rightarrow Y_1 \). Middle row shows the results of consistent training of \( x \rightarrow Y_1 \) by leveraging \( Y_1 \rightarrow Y \) in the loss. The predicted normals are notably improved, especially in hard to predict fine-grained details (zoom into the yellow markers. Best seen on screen). Bottom row provides the ground truth. See video examples at visualizations webpage.

**Figure 5:** Schematic summary of derived losses for \( f_{XY_1} \). (a): \( \mathcal{L}_{\text{perceptual}}^{\text{XY_1}} \) (Eq.1). (b): \( \mathcal{L}_{\text{perceptual}}^{\text{XY_2}} \) (Eq.3). (c): \( \mathcal{L}_{\text{perceptual}}^{\text{XY_2}} \) (Eq.6). (d): \( \mathcal{L}_{\text{perceptual}}^{\text{XY_1}} \) (Eq.7).
triangle-based units. We also assumed the functions $F_y$ were given apriori. The more general multi-task setup is: given a large set of domains, we are interested in learning functions that map the domains onto each other in a globally cross-task consistent manner. This objective can be formulated over a graph $G = (D, F)$ with nodes representing all of the domains $D = \{x \cup y\}$ and edges being neural networks between them $F = (F_x \cup F_y)$; see Fig. 2(c).

**Extension to Arbitrary Paths:** The transition from three domains to a large graph $G$ enables forming more general consistency constraints using arbitrary-paths. That is, two paths with same endpoint should yield the same results—an example is shown in Fig. 2(d). The triangle constraint in Fig. 2(b,c) is a special case of the more general constraint in Fig. 2(d), if paths with lengths 1 and 2 are picked for the green and blue paths. Extending the derivations done for a triangle in Sec. 3.1 to paths yields:

$$L_{\text{perceptual}}^{\text{cons}} = |f_{xy_1}(x) - y_1| + |f_{y_k - y_{k+1} \circ \ldots \circ y_1}(x) - f_{y_k - y_{k+1} \circ \ldots \circ y_1}(y_1)|,$$

which is the loss for training $f_{xy_1}$ using the arbitrary consistency path $x \rightarrow y_1 \rightarrow y_2 \rightarrow \ldots \rightarrow y_k$ with length $k$ (full derivation provided in supplementary material). Notice that Eq. 6 is a special case of Eq. 8 if $k=2$. Equation 8 is particularly useful for incomplete graphs; if the function $y_1 \rightarrow y_k$ is missing, consistency between domains $y_1$ and $y_k$ can still be enforced via transitivity through other domains using Eq. 8.

Also, extending Eq. 8 to multiple simultaneous paths (as in Eq. 7) by summing the path constraints is straightforward.

**Global Consistency Objective:** We define reaching global cross-task consistency for graph $G$ as satisfying the consistency constraint for all feasible paths in $G$. We can write the global consistency objective for $G$ as $L_{\text{G}} = \sum_{p \in P} L_{p}^{\text{perceptual}}$, where $p$ represents a path and $P$ is the set of all feasible paths in $G$.

Optimizing the objective $L_{\text{G}}$ directly is intractable as it would require simultaneous training of all networks in $F$ with a massive number of consistency paths. In Alg. 1 we devise a straightforward training schedule for an approximate optimization of $L_{\text{G}}$. This problem is similar to inference in graphical models, where one is interested in marginal distribution of unobserved nodes given some observed nodes by passing “messages” between them through the graph until convergence. As exact inference is usually intractable for unconstrained graphs, often an approximate message passing algorithm with various heuristics is used.

Algorithm 1 selects one network $f_{ij} \in F$ to be trained, selects consistency path(s) $p \in P$ for it, and trains $f_{ij}$ with $p$ for a fixed number of steps using loss 8 (or its multi path version if multiple paths selected). This is repeated until all networks in $F$ satisfy a convergence criterion.

A number of choices for the selection criterion in SelectNetwork and SelectPath is possible, including round-robin and random selection. While we did not observe a significant difference in the final results, we achieved the best results using maximal violation criterion: at each step select the network and path with the largest loss. Also, Alg. 1 starts from shorter paths and progressively opens up to longer ones (up to length $L$) only after shorter paths have converged. This is based on the observation that the benefit of short and long paths in terms of enforcing cross-task consistency overlap, while shorter paths are computationally cheaper.

For the same reason, all of the networks are initialized by training using the standard direct loss (Op. 1 in Alg. 1) before progressively adding consistency terms.

Finally, Alg. 1 does not distinguish between $F_x$ and $F_y$ and can be used to train them all in the same pool. This means the selected path $p$ may include networks not fully converged yet. This is not an issue in practice, because, first, all networks are pre-trained with their direct loss (Op. 1 in Alg. 1) thus they are not wildly far from their convergence point. Second, the perceptual loss formulation makes training $f_{ij}$ robust to imperfections in functions in $p$ (Sec. 3.1.2). However, as practical applications primarily care about $F_x$, rather than $F_y$, one can first train $F_y$ to convergence using Alg. 1, then start the training of $F_x$ with well trained and converged networks $F_y$. We do the latter in our experiments.

Please see supplementary material for how to normalize and balance the direct and consistency loss terms, as they belong to different domains with distinct numerical properties.

**4. Consistency Energy**

We quantify the amount of cross-task consistency in the system using an energy-based quantity [26] called Consistency Energy. For a single query $x$ and domain $y_k$, the consistency energy is defined to be the standardized average of pairwise inconsistencies:

$$\text{Energy}_{y_k}(x) = \frac{1}{|P| - 1} \sum_{y_{1} \neq y_{k}, y \neq k} \frac{|f_{y_{1}y_{k}} \circ f_{xy_{1}}(x) - f_{xy_{1}}(x)|}{\mu_{i}},$$

where $\mu_{i}$ and $\sigma_{i}$ are the average and standard deviation of $|f_{y_{1}y_{k}} \circ f_{xy_{1}}(x) - f_{xy_{1}}(x)|$ over the dataset. Eq. 9 can be computed per-pixel or per-image by average over its pixels. Intuitively, the energy can be thought of as the amount of
5. Experiments

The evaluations are organized to demonstrate the proposed approach yields predictions that are I. more consistent (Sec.5.1), II. more accurate (Sec.5.2), and III. more generalizable to out-of-training-distribution data (Sec.5.4). We also IV. quantitatively analyze the Consistency Energy and report its utilities (Sec.5.3).

Datasets: We used the following datasets in the evaluations:

Taskonomy [45]: We adopted Taskonomy as our main training dataset. It includes 4 million real images of indoor scenes with multi-task annotations for each image. The experiments were performed using the following 10 domains from the dataset: RGB images, surface normals, principal curvature, depth (z-buffer), reshading, 3D (occlusion) edges, 2D (Sobel) texture edges, 3D keypoints, 2D keypoints, semantic segmentation. The tasks were selected to cover 2D, 3D, and semantic domains and have sensor-based/semantic ground truth. We report results on the test set.

Replica[40] has high resolution 3D ground truth and enables more reliable evaluations of fine-grained details. We test on 1227 images from Replica (no training), besides Taskonomy test data.

CocoDoom [27] contains synthetic images from the Doom video game. We use it as one of the out-of-training-distribution datasets.

ApolloScape [16] contains real images of outdoor driving scenes. We use it as another out-of-training-distribution dataset.

NYU [37]: We also evaluated on NYUv2. The findings are similar to those on Taskonomy and Replica (in supplementary material).

Architecture & Training Details: We used a UNet [34] backbone architecture. All networks in $\mathcal{F}_x$ and $\mathcal{F}_y$ have a similar architecture. The networks have 6 down and 6 up sampling blocks and were trained using AMSGrad [33] and Group Norm [42] with learning rate $3 \times 10^{-5}$, weight decay $2 \times 10^{-6}$, and batch size 32. Input and output images were linearly scaled to the range $[0, 1]$ and resized down to $256 \times 256$. We used $\ell_1$ as the norm in all losses and set the max path length $L=3$.

Baselines: The main baseline categories are described below. To prevent confounding factors, our method and all baselines were implemented using the same UNet network when feasible and were re-trained on Taskonomy dataset.

Baseline UNet (standard independent learning) is the main baseline. It is identical to consistency models in all senses, except being trained with only the direct loss and no consistency terms.

Multi-task learning: A network with one shared encoder and multiple decoders each dedicated to a task, similar to [21].

Cycle-based consistency, e.g.[50], is a way of enforcing consistency requiring a bijection between domains. This baseline is the special case of the triangle in Fig.2(b) by setting $y_2 = x$.

Baseline perceptual loss network uses frozen random (Gaussian weight) networks as $\mathcal{F}_y$, rather than training them to be cross-task functions. This baseline would show if the improvements were owed to the priors in the architecture of constraint networks, rather than them executing cross-task consistency constraints.

GAN-based image translation: We used Pix2Pix [17].
5. Consistency of Predictions

Fig. 9(a) (blue) shows the amount of inconsistency in test set predictions (Consistency Energy) successfully decreases over the course of training. The convergence point of the network trained with consistency is well below baseline independent learning (orange) and multi-task learning (green)—which shows consistency among predictions does not naturally emerge in either case without explicit constraining.

5.2. Accuracy of Predictions

Figures 6 and 7 compare the prediction results of networks trained with cross-task consistency against the baselines in different domains. The improvements are considerable particularly around the difficult fine-grained details.

Quantitative evaluations are provided in Tab. 1 for Replica dataset and Taskonomy datasets on depth, normal, reshading, and pixel-wise semantic prediction tasks. Learning with consistency led to large improvements in most of the setups. As most of the pixels in an image belong to easy to predict regions governed by the room layout (e.g., ceiling, walls), the standard pixel-wise error metrics (e.g. $\ell_1$) are dominated by them and consequently insensitive to fine-grained changes. Thus, besides standard Direct metrics, we report Perceptual error metric (e.g. normal=curvature) that evaluate the same prediction, but with a non-uniform attention to pixel properties. Each perceptual error provides a different angle, and the optimal results would have a low error for all metrics.

Table 1: Quantitative Evaluation of Cross-Task Consistent Learning vs Baselines. Results are reported on Replica and Taskonomy Datasets for four prediction tasks (normal, depth, reshading, pixel-wise semantic labeling) using 'Direct' and 'Perceptual' error metrics. The Perceptual metrics evaluate the target prediction in another domain (e.g., the leftmost column evaluates the depth inferred out of the predicted normals). Bold marks the best-performing method. If more than one value is bold, their performances were statistically indistinguishable from the best, according to 2-sample paired t-test $\alpha = 0.01$. Learning with consistency led to improvements with large margins in most columns. (In all tables, $\ell$ norm values are multiplied by 100 for readability. Methods that cannot be run for a given target are denoted by ‘×’.)

Blind guess: A query-agnostic statistically informed guess computed from data for each domain (visually in supplementary). It shows what can be learned from general dataset regularities. [45] GeoNet [32] is a task-specific consistency method analytically curated for depth and normals. This baseline shows how closely the task-specific consistency methods based on known analytical relationships perform vs the proposed generic data-driven method. The “original” and “updated” variants represent original authors’ released networks and our re-implemented and re-trained version.

5.3. Utilities of Consistency Energy

Below we quantitatively analyze the Consistency Energy. The energy is shown (per-pixel) for sample queries in Fig. 6.

Consistency Energy as a Confidence Metric: The plot 9(b) shows the energy of predictions has a strong positive correlation with the error computed using ground truth (Pearson corr. 0.67). This suggests the energy can be adopted for confidence quantification and handling uncertainty. This experiment was done on Taskonomy test set.

Blind guess: A query-agnostic statistically informed guess computed from data for each domain (visually in supplementary). It shows what can be learned from general dataset regularities. [45] GeoNet [32] is a task-specific consistency method analytically curated for depth and normals. This baseline shows how closely the task-specific consistency methods based on known analytical relationships perform vs the proposed generic data-driven method. The “original” and “updated” variants represent original authors’ released networks and our re-implemented and re-trained version.

5.3. Utilities of Consistency Energy

Below we quantitatively analyze the Consistency Energy. The energy is shown (per-pixel) for sample queries in Fig. 6.

Consistency Energy as a Confidence Metric: The plot 9(b) shows the energy of predictions has a strong positive correlation with the error computed using ground truth (Pearson corr. 0.67). This suggests the energy can be adopted for confidence quantification and handling uncertainty. This experiment was done on Taskonomy test set.

Blind guess: A query-agnostic statistically informed guess computed from data for each domain (visually in supplementary). It shows what can be learned from general dataset regularities. [45] GeoNet [32] is a task-specific consistency method analytically curated for depth and normals. This baseline shows how closely the task-specific consistency methods based on known analytical relationships perform vs the proposed generic data-driven method. The “original” and “updated” variants represent original authors’ released networks and our re-implemented and re-trained version.

5.3. Utilities of Consistency Energy

Below we quantitatively analyze the Consistency Energy. The energy is shown (per-pixel) for sample queries in Fig. 6.

Consistency Energy as a Confidence Metric: The plot 9(b) shows the energy of predictions has a strong positive correlation with the error computed using ground truth (Pearson corr. 0.67). This suggests the energy can be adopted for confidence quantification and handling uncertainty. This experiment was done on Taskonomy test set.
Figure 8: Error with Increasing (Smooth) Domain Shift. The network trained with consistency is more robust to the shift.

Figure 9: Analyses of Consistency Energy.

Consistency Energy as a Domain Shift Detector: Plot 9(c) shows the energy distribution of in-distribution (Taskonomy) and out-of-distribution datasets (ApolloScape, CocoDoom). Out-of-distribution datapoints have notably higher energy values, which suggests that energy can be used to detect anomalous samples or domain shifts. Using the per-image energy value to detect out-of-distribution images achieved ROC-AUC=0.95; the out-of-distribution detection method OC-NN [3] scored 0.51.

Plot 9(d) shows the same concept as 9(c) (energy vs domain shift), but when the shift away from the training data is smooth. The shift was done by applying a progressively stronger Gaussian blur with kernel size 6 on Taskonomy test images. The plot also shows the error computed using ground truth which has a pattern similar to the energy.

We find the reported utilities noteworthy as handling uncertainty, domains shifts, and measuring prediction confidence in neutral networks are open topics of research [29, 11] with critical values in, e.g. active learning [35], real-world decision making [20], and robotics [31].

5.4. Generalization & Adaptation to New Domains

To study: I. how well the networks generalize to new domains without any adaptation and quantify their resilience, and II. how efficiently they can adapt to a new domain given a few training examples by fine-tuning, we test the networks trained on Taskonomy dataset on various new domains.

In interest of space, we defer the details to the supplementary and provide the results in Fig. 8 and Tab. 2. Networks trained with consistency generally show higher resilience w.r.t. domain shifts and better adaptation with little data.

Supplementary Material: We defer additional discussions and experiments, particularly analyzing different aspects of the optimization, stability analysis of the experimental trends, and proving qualitative results at scale to the supplementary material and the project page.

6. Conclusion and Limitations

We presented a general and data-driven framework for augmenting standard learning with cross-task consistency. The evaluations showed learning with cross-task consistency fits the data better yielding more accurate predictions and leads to models with improved generalization. The Consistency Energy was found to be an informative intrinsic quantity with utilities toward confidence estimation and domain shift detection. We briefly discuss some of the limitations:

Path Ensembles: We used the various inference paths only as a way of enforcing consistency. Aggregation of multiple (comparably weak) inference paths into a single strong estimator (e.g. in a manner similar to boosting) is a promising direction that this paper did not address.

Categorical/Low-Dimensional Tasks: We primarily experimented with pixel-wise tasks. Classification tasks, and generally tasks with low-dimensional outputs, will be interesting to experiment with, especially given the more severely ill-posed cross-task relationships they induce.

Unlabeled/Unpaired Data: The current framework requires labeled training data. Extending the concept to unlabeled/unpaired data, e.g. as in [50], remains open.

Optimization Limits: The improvements gained by incorporating consistency are bounded by the success of the available optimization techniques, as addition of consistency constrains at times makes the optimization job harder. Also, implementing cross-task functions using neural networks makes them subject to certain output artifacts similar to those seen in image synthesis with neural networks.

Adversarial Robustness: Lastly, if learning with cross-task consistency indeed reduces the tendency of neural networks to learn surface statistics [18] (Sec. 1), studying its implications in defenses against adversarial attacks will be valuable.

Acknowledgement: We thank Teresa Yeo and Oğuzhan Kar. This work was supported by ONR MURI (N00014-14-1-0671), an Amazon AWS Machine Learning Award, Google Cloud, and TRI. Toyota Research Institute (“TRI”) provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

<table>
<thead>
<tr>
<th>Novel Domain</th>
<th>Error (Post-Adaptation)</th>
<th>Error (Pre-Adaptation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian blur</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taskonomy</td>
<td>128</td>
<td>14 (+14.7%)</td>
</tr>
<tr>
<td>ApolloScape</td>
<td>8</td>
<td>40.5 (+11.9%)</td>
</tr>
<tr>
<td>CocoDoom</td>
<td>128</td>
<td>27.1 (+24.5%)</td>
</tr>
</tbody>
</table>

Table 2: Domain generalization and adaptation on CocoDoom, ApolloScape, and Taskonomy blur data. Networks trained with consistency show better generalization to new domains and a faster adaptation with little data. (relative improvement in parentheses)
References


[13] Steven Hickson, Karthik Raveendran, Alireza Fathi, Kevin Murphy, and Irfan Essa. Floors are flat: Leveraging semantics for real-time surface normal prediction. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 0–0, 2019. 2


[29] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift, 2019. 8


