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****.[1] 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

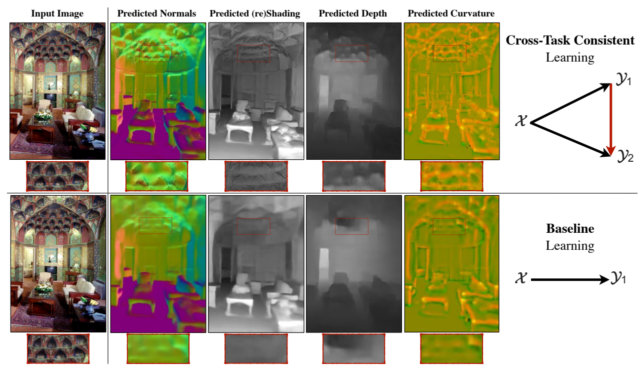


Figure 1: **Cross-Task Consistent Learning.** The predictions made for different tasks out of one image are expected to be *consistent*, as the underlying scene is the same. This is exemplified by a challenging query and four sample predictions out of it. We propose a general method for learning utilizing data-driven cross-task consistency constraints. The lower and upper rows show the results of the baseline (independent learning) and learning with consistency, which yields higher quality and more consistent predictions. Red boxes provide magnifications. [Best seen on screen]

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.[2] 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

[1] Abbreviated **X-TC**, standing for **Cross-T**ask **C**onsistency.
*Equal.

[2] For 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, 8, 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

[3]inference-path invariance was inspired by **Conservative Vector Fields** in vector calculus and physics that are (at a high level) fields in which integration along *different paths yield the same results, as long as their endpoints are the same* [10]. Many key concepts in physics are 'conservative', e.g., gravitational force: the work done against gravity when moving between two points is independent of the path taken.

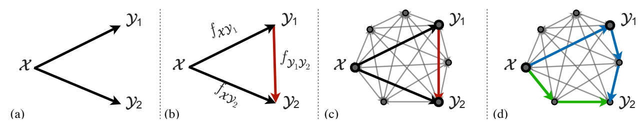


Figure 2: **Enforcing Cross-Task Consistency: (a)** shows the typical multitask setup where predictions $\mathcal{X}\rightarrow\mathcal{Y}_1$ and $\mathcal{X}\rightarrow\mathcal{Y}_2$ are trained without a notation of consistency. **(b)** depicts the elementary *triangle consistency constraint* where the prediction $\mathcal{X}\rightarrow\mathcal{Y}_1$ is enforced to be consistent with $\mathcal{X}\rightarrow\mathcal{Y}_2$ using a function that relates $\mathcal{Y}_1$ to $\mathcal{Y}_2$ (i.e. $\mathcal{Y}_1\rightarrow\mathcal{Y}_2$). **(c)** shows how the triangle unit from (b) can be an element of a larger system of domains. Finally, **(d)** illustrates the generalized case where in the larger system of domains, consistency can be enforced using invariance along arbitrary paths, as long as their endpoints are the same (here the blue and green paths). This is the general concept behind *inference-path invariance*. The triangle in (b) is the smallest unit of such paths.

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 $\mathcal{X}$ denotes the query domain (e.g., RGB images) and $\mathcal{Y}=\{\mathcal{Y}_1,...,\mathcal{Y}_n\}$ is the set of n desired prediction domains (e.g. normals, depth, objects, etc). An individual datapoint from domains $(\mathcal{X},\mathcal{Y}_1,...,\mathcal{Y}_n)$ 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. $\mathcal{F}_\mathcal{X}=\{f_{\mathcal{X}\mathcal{Y}_j}|\mathcal{Y}_j\in\mathcal{Y}\}$ where $f_{\mathcal{X}\mathcal{Y}_j}(x)$ outputs y_j given x. We also define $\mathcal{F}_\mathcal{Y}=\{f_{\mathcal{Y}_i\mathcal{Y}_j}|\mathcal{Y}_i,\mathcal{Y}_j\in\mathcal{Y}, i\neq j\}$, 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 $\mathcal{F}_\mathcal{Y}$ is given apriori and frozen; in Sec. 3.3 we discuss all functions fs are neural networks in this paper, and we learn $\mathcal{F}_\mathcal{Y}$ just like $\mathcal{F}_\mathcal{X}$.

3.1. Triangle: The Elementary Consistency Unit

The typical supervised way of training the neural networks in $\mathcal{F}_\mathcal{X}$, e.g. $f_{\mathcal{X}\mathcal{Y}_1}(x)$, is to find parameters of $f_{\mathcal{X}\mathcal{Y}_1}$ that minimize a loss with the general form $|f_{\mathcal{X}\mathcal{Y}_1}(x)\text{-}y_1|$ using a

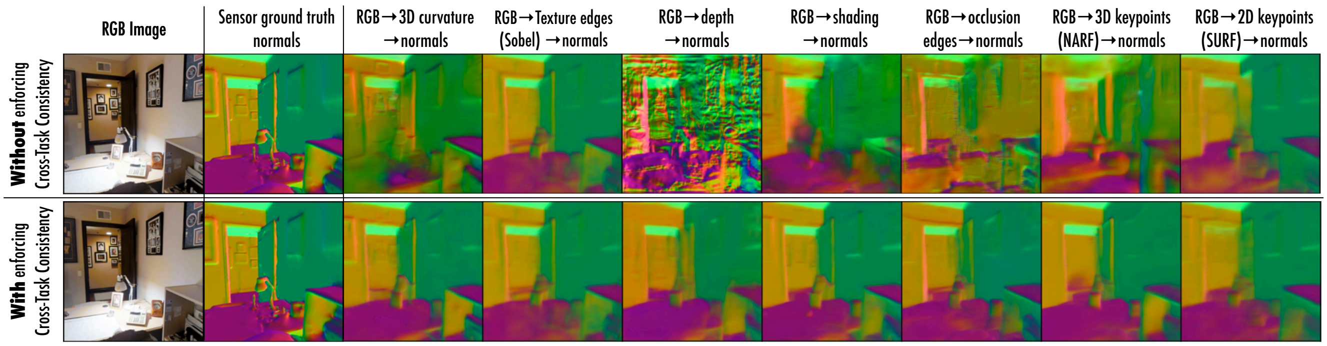


Figure 3: **Impact of disregarding cross-task consistency in learning**, illustrated using surface normals domain. Each subfigure shows the results of predicting surface normals out of the prediction of an intermediate domain; using the notation $\mathcal{X}\rightarrow\mathcal{Y}_1\rightarrow\mathcal{Y}_2$, here $\mathcal{X}$ is RGB image, $\mathcal{Y}_2$ is surface normals, and each column represents a different $\mathcal{Y}_1$. The **upper row** demonstrates the normals are noisy and dissimilar when cross-task consistency is not incorporated in learning of $\mathcal{X}\rightarrow\mathcal{Y}_1$ networks. Whereas enforcing consistency when learning $\mathcal{X}\rightarrow\mathcal{Y}_1$ 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.

distance function as $|.|$, e.g. ℓ_1 norm. This standard *independent* learning of $f_{\mathcal{X}\mathcal{Y}_i}$s 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 $\mathcal{Y}_1$ from $\mathcal{X}$ *while enforcing consistency with domain* $\mathcal{Y}_2$ as a directed triangle depicted in Fig. 2(b):

$$\mathcal{L}^{triangle}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2} \triangleq |f_{\mathcal{X}\mathcal{Y}_1}(x)\text{-}y_1| + |f_{\mathcal{Y}_1\mathcal{Y}_2}\circ f_{\mathcal{X}\mathcal{Y}_1}(x)\text{-}f_{\mathcal{X}\mathcal{Y}_2}(x)| + |f_{\mathcal{X}\mathcal{Y}_2}(x)\text{-}y_2|. \tag{1}$$

The first and last terms are the standard *direct* losses for training $f_{\mathcal{X}\mathcal{Y}_1}$ and $f_{\mathcal{X}\mathcal{Y}_2}$. The middle term is the *consistency term* which enforces predicting $\mathcal{Y}_2$ out of the predicted $\mathcal{Y}_1$ yields the same result as directly predicting $\mathcal{Y}_2$ out of $\mathcal{X}$.[4] Thus learning to predict $\mathcal{Y}_1$ and $\mathcal{Y}_2$ 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 $\mathcal{L}^{triangle}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ involves *simultaneous* training of two networks $f_{\mathcal{X}\mathcal{Y}_1}$ and $f_{\mathcal{X}\mathcal{Y}_2}$, thus it is resource demanding. We show $\mathcal{L}^{triangle}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ can be reduced to a 'separable' function [39] resulting in two terms that can be optimized independently.

From triangle inequality we can derive:

$$|f_{\mathcal{Y}_1\mathcal{Y}_2}\circ f_{\mathcal{X}\mathcal{Y}_1}(x)\text{-}f_{\mathcal{X}\mathcal{Y}_2}(x)| \leq |f_{\mathcal{Y}_1\mathcal{Y}_2}\circ f_{\mathcal{X}\mathcal{Y}_1}(x)\text{-}y_2| + |f_{\mathcal{X}\mathcal{Y}_2}(x)\text{-}y_2|,$$

which after substitution in Eq. 1 yields:

$$\mathcal{L}^{triangle}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2} \leq |f_{\mathcal{X}\mathcal{Y}_1}(x)\text{-}y_1| + |f_{\mathcal{Y}_1\mathcal{Y}_2}\circ f_{\mathcal{X}\mathcal{Y}_1}(x)\text{-}y_2| + 2|f_{\mathcal{X}\mathcal{Y}_2}(x)\text{-}y_2|. \tag{2}$$

The upper bound for $\mathcal{L}^{triangle}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ in inequality 2 can be optimized in lieu of $\mathcal{L}^{triangle}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ itself, as they both have the same minimizer.[5] The terms of this bound include either $f_{\mathcal{X}\mathcal{Y}_1}$ or $f_{\mathcal{X}\mathcal{Y}_2}$, but not both, hence we now have a loss separable into functions of $f_{\mathcal{X}\mathcal{Y}_1}$ or $f_{\mathcal{X}\mathcal{Y}_2}$, and they can be optimized independently. The part pertinent to the network $f_{\mathcal{X}\mathcal{Y}_1}$ is:

$$\mathcal{L}^{separate}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2} \triangleq |f_{\mathcal{X}\mathcal{Y}_1}(x) - y_1| + |f_{\mathcal{Y}_1\mathcal{Y}_2}\circ f_{\mathcal{X}\mathcal{Y}_1}(x) - y_2|, \tag{3}$$

named *separate*, as we reduced the closed triangle objective $\mathcal{X}\overset{\mathcal{Y}_1}{\triangle}\mathcal{Y}_2$ in Eq. 1 to two equivalent separate path objectives $\mathcal{X}\rightarrow\mathcal{Y}_1\rightarrow\mathcal{Y}_2$ and $\mathcal{X}\rightarrow\mathcal{Y}_2$. The first term of Eq. 3 enforces the general correctness of predicting $\mathcal{Y}_1$, and the second term enforces its consistency with $\mathcal{Y}_2$ domain.

3.1.2 Reconfiguration into a "Perceptual Loss"

Training $f_{\mathcal{X}\mathcal{Y}_1}$ using the loss $\mathcal{L}^{separate}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ 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 $f_{\mathcal{Y}_1\mathcal{Y}_2}$ for mapping $\mathcal{Y}_1$ onto $\mathcal{Y}_2$; i.e. it demands $y_2{=}f_{\mathcal{Y}_1\mathcal{Y}_2}(y_1)$. We show how these two requirements can be reduced.

Again, from triangle inequality we can derive:

$$|f_{\mathcal{Y}_1\mathcal{Y}_2}\circ f_{\mathcal{X}\mathcal{Y}_1}(x) - y_2| \leq |f_{\mathcal{Y}_1\mathcal{Y}_2}\circ f_{\mathcal{X}\mathcal{Y}_1}(x) - f_{\mathcal{Y}_1\mathcal{Y}_2}(y_1)| + |f_{\mathcal{Y}_1\mathcal{Y}_2}(y_1) - y_2|, \tag{4}$$

which after substitution in Eq. 3 yields:

$$\mathcal{L}^{separate}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2} \leq |f_{\mathcal{X}\mathcal{Y}_1}(x) - y_1| + |f_{\mathcal{Y}_1\mathcal{Y}_2}\circ f_{\mathcal{X}\mathcal{Y}_1}(x) - f_{\mathcal{Y}_1\mathcal{Y}_2}(y_1)| + |f_{\mathcal{Y}_1\mathcal{Y}_2}(y_1) - y_2|. \tag{5}$$

Similar to the discussion for inequality 2, the upper bound in inequality 5 can be optimized in lieu of $\mathcal{L}^{separate}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ as both have the same minimizer.[6] As the last term is a constant w.r.t.

[4] Operator $\circ$ denotes function composition: $g\circ h(x) \triangleq g(h(x))$.

[5] Both sides of inequality 2 are ≥ 0 and $=0$ for the minimizer $f_{\mathcal{X}\mathcal{Y}_1}(x){=}y_1$ & $f_{\mathcal{X}\mathcal{Y}_2}(x){=}y_2$.

[6] Both sides of inequality 5 are ≥ 0 and $=0$ for the minimizer $f_{\mathcal{X}\mathcal{Y}_1}(x){=}y_1$. The term $|f_{\mathcal{Y}_1\mathcal{Y}_2}(y_1) - y_2|$ is a constant and ~ 0, as it is exactly the training objective of $f_{\mathcal{Y}_1\mathcal{Y}_2}$. The non-zero residual should be ignored and assumed 0 as the non-zero part is irrelevant to $f_{\mathcal{X}\mathcal{Y}_1}$, but imperfections of $f_{\mathcal{Y}_1\mathcal{Y}_2}$.

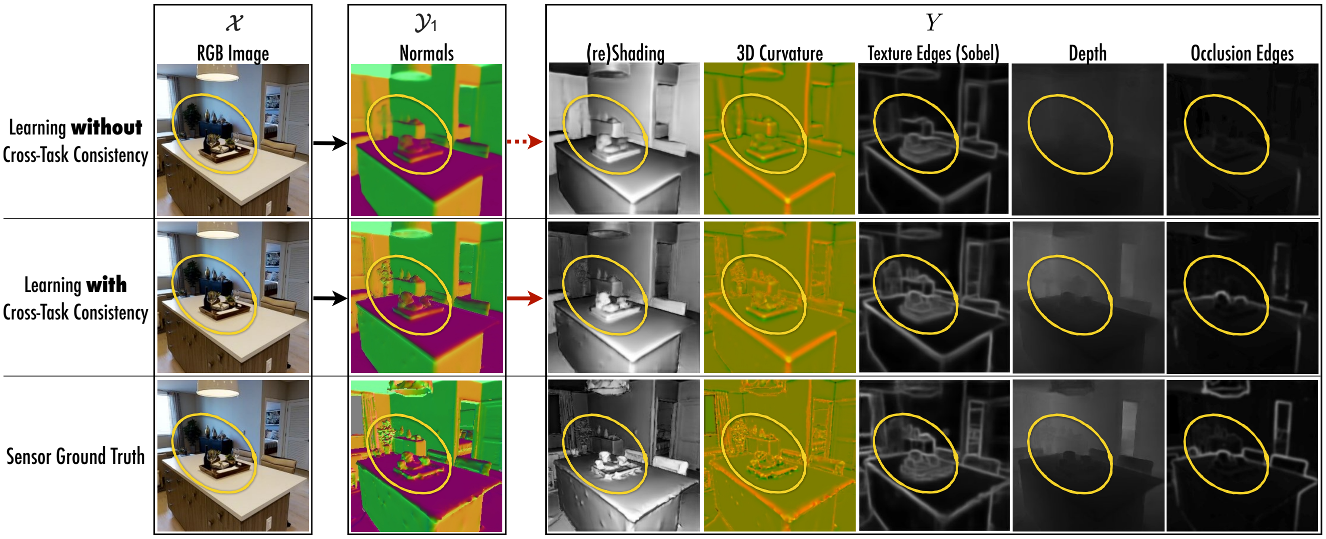


Figure 4: **Learning with and without cross-task consistency shown for a sample query**. Using the notation $\mathcal{X}\rightarrow\mathcal{Y}_1\rightarrow Y$, here $\mathcal{X}$ is RGB image, $\mathcal{Y}_1$ is surface normals, and five domains in Y are reshading, 3D curvature, texture edges (Sobel filter), depth, and occlusion edges.
Top row shows the results of standard training of $\mathcal{X}\rightarrow\mathcal{Y}_1$. After convergence of training, the predicted normals ($\mathcal{Y}_1$) are projected onto other domains (Y) which reveal various inaccuracies. This demonstrates such cross-task projections $\mathcal{Y}_1\rightarrow Y$ can provide additional cues to training $\mathcal{X}\rightarrow\mathcal{Y}_1$.
Middle row shows the results of consistent training of $\mathcal{X}\rightarrow\mathcal{Y}_1$ by leveraging $\mathcal{Y}_1\rightarrow Y$ 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.

$f_{\mathcal{X}\mathcal{Y}_1}$, the final loss for training $f_{\mathcal{X}\mathcal{Y}_1}$ is:

$$\mathcal{L}^{perceptual}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2} \triangleq |f_{\mathcal{X}\mathcal{Y}_1}(x)-y_1| + |f_{\mathcal{Y}_1\mathcal{Y}_2}\circ f_{\mathcal{X}\mathcal{Y}_1}(x)-f_{\mathcal{Y}_1\mathcal{Y}_2}(y_1)|. \quad (6)$$

The loss $\mathcal{L}^{perceptual}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ no longer includes y_2, hence it admits pair training data (x, y_1) rather than triplet (x, y_1, y_2). Comparing $\mathcal{L}^{perceptual}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ and $\mathcal{L}^{separate}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ shows the modification boiled down to replacing y_2 with $f_{\mathcal{Y}_1\mathcal{Y}_2}(y_1)$. 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 $f_{\mathcal{Y}_1\mathcal{Y}_2}$ is a *noisy* estimator, then $f_{\mathcal{Y}_1\mathcal{Y}_2}(y_1){=}y_2{+}noise$ rather than $f_{\mathcal{Y}_1\mathcal{Y}_2}(y_1){=}y_2$. Using a noisy $f_{\mathcal{Y}_1\mathcal{Y}_2}$ in $\mathcal{L}^{separate}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ corrupts the training of $f_{\mathcal{X}\mathcal{Y}_1}$ since the second loss term does not reach 0 if $f_{\mathcal{X}\mathcal{Y}_1}(x)$ correctly outputs y_1. That is in contrast to $\mathcal{L}^{perceptual}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ where both terms have the same global minimum and are always 0 if $f_{\mathcal{X}\mathcal{Y}_1}(x)$ outputs y_1 – even when $f_{\mathcal{Y}_1\mathcal{Y}_2}(y_1){=}y_2{+}noise$. 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 $\mathcal{Y}_1$ and $\mathcal{Y}_2$ in Eq. 6 (second term) can be viewed as judging the prediction $f_{\mathcal{X}\mathcal{Y}_1}(x)$ against y_1 via the lens of the network $f_{\mathcal{Y}_1\mathcal{Y}_2}$;

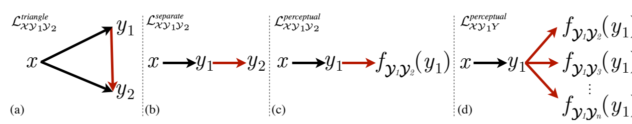


Figure 5: **Schematic summary of derived losses for** $f_{\mathcal{X}\mathcal{Y}_1}$ **.(a)**: $\mathcal{L}^{triangle}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ (Eq.1). **(b)**: $\mathcal{L}^{separate}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ (Eq.3). **(c)**: $\mathcal{L}^{perceptual}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ (Eq.6). **(d)**: $\mathcal{L}^{perceptual}_{\mathcal{X}\mathcal{Y}_1 Y}$ (Eq.7).

here $f_{\mathcal{Y}_1\mathcal{Y}_2}$ is a "perceptual loss" for training $f_{\mathcal{X}\mathcal{Y}_1}$. However, unlike the ImageNet-based perceptual loss [19], this function has the specific and interpretable job of enforcing consistency with another task. We also use multiple $f_{\mathcal{Y}_1\mathcal{Y}_i}$s simultaneously which enforces consistency of predicting $\mathcal{Y}_1$ against multiple other domains (Sections 3.2 and 3.3).

3.2. Consistency of $f_{\mathcal{X}\mathcal{Y}_1}$ with 'Multiple' Domains

The derived $\mathcal{L}^{perceptual}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ loss augments learning of $f_{\mathcal{X}\mathcal{Y}_1}$ with a consistency constraint against *one* domain $\mathcal{Y}_2$. Straightforward extension of the same derivation to enforcing consistency of $f_{\mathcal{X}\mathcal{Y}_1}$ against *multiple* other domains (i.e. when $f_{\mathcal{X}\mathcal{Y}_1}$ is part of multiple simultaneous triangles) yields:

$$\mathcal{L}^{perceptual}_{\mathcal{X}\mathcal{Y}_1 Y} \triangleq |Y|{\times}|f_{\mathcal{X}\mathcal{Y}_1}(x)\text{-}y_1| + \sum_{\mathcal{Y}_i\in Y} |f_{\mathcal{Y}_1\mathcal{Y}_i}\circ f_{\mathcal{X}\mathcal{Y}_1}(x)\text{-}f_{\mathcal{Y}_1\mathcal{Y}_i}(y_1)|, \quad (7)$$

where Y is the *set* of domains with which $f_{\mathcal{X}\mathcal{Y}_1}$ must be consistent, and $|Y|$ is the cardinality of Y. Notice that $\mathcal{L}^{perceptual}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2}$ is a special case of $\mathcal{L}^{perceptual}_{\mathcal{X}\mathcal{Y}_1 Y}$ where $Y{=}\{\mathcal{Y}_2\}$. Fig. 5 summarizes the derivation of losses for $f_{\mathcal{X}\mathcal{Y}_1}$.

Fig. 4 shows qualitative results of learning $f_{\mathcal{X}\mathcal{Y}_1}$ 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 $f_{\mathcal{X}\mathcal{Y}_1}$ using elementary *triangle* based units. We also assumed the functions $\mathcal{F}_{\mathcal{Y}}$ 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 $\mathcal{G}{=}(\mathcal{D}, \mathcal{F})$ with nodes representing all of the domains $\mathcal{D}{=}(\mathcal{X} \cup \mathcal{Y})$ and edges being neural networks between them $\mathcal{F}{=}(\mathcal{F}_{\mathcal{X}} \cup \mathcal{F}_{\mathcal{Y}})$; see Fig.2(c).

Extension to Arbitrary Paths: The transition from three domains to a large graph $\mathcal{G}$ 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:

$$\mathcal{L}^{\textit{perceptual}}_{\mathcal{X}\mathcal{Y}_1\mathcal{Y}_2..\mathcal{Y}_k} = |f_{\mathcal{X}\mathcal{Y}_1}(x) - y_1| + \\ |f_{\mathcal{Y}_{k-1}\mathcal{Y}_k} \circ ... \circ f_{\mathcal{Y}_1\mathcal{Y}_2} \circ f_{\mathcal{X}\mathcal{Y}_1}(x) - f_{\mathcal{Y}_{k-1}\mathcal{Y}_k} \circ ... \circ f_{\mathcal{Y}_1\mathcal{Y}_2}(y_1)|, \quad (8)$$

which is the loss for training $f_{\mathcal{X}\mathcal{Y}_1}$ using the arbitrary consistency path $\mathcal{X} \rightarrow \mathcal{Y}_1 \rightarrow \mathcal{Y}_2 ... \rightarrow \mathcal{Y}_k$ 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 $\mathcal{Y}_1 \rightarrow \mathcal{Y}_k$ is missing, consistency between domains $\mathcal{Y}_1$ and $\mathcal{Y}_k$ 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 $\mathcal{G}$ as satisfying the consistency constraint for *all* feasible paths in $\mathcal{G}$. We can write the global consistency objective for $\mathcal{G}$ as $\mathcal{L}_{\mathcal{G}} = \sum_{p \in \mathcal{P}} \mathcal{L}^{\textit{perceptual}}_{p}$, where p represents a path and $\mathcal{P}$ is the set of all feasible paths in $\mathcal{G}$.

Optimizing the objective $\mathcal{L}_{\mathcal{G}}$ directly is intractable as it would require simultaneous training of all networks in $\mathcal{F}$ with a massive number of consistency paths[7]. In Alg.1 we devise a straightforward training schedule for an approximate optimization of $\mathcal{L}_{\mathcal{G}}$. 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 $f_{ij} {\in} \mathcal{F}$ to be trained, selects consistency path(s) $p {\in} \mathcal{P}$ 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 $\mathcal{F}$ satisfy a convergence criterion.

[7]For example, a complete $\mathcal{G}$ with n nodes includes $n(n-1)$ networks and $\sum_{k=2}^{L} \binom{n}{k+1}(k+1)!$ feasible paths, with path length capped at L.

Algorithm 1: Globally Cross-Task Consistent Learning of Networks $\mathcal{F}$

Result: Trained edges $\mathcal{F}$ of graph $\mathcal{G}$
1 Train each $f {\in} \mathcal{F}$ independently. ▷ initialization by standard direct training.
2 **for** $k \leftarrow 2$ *to* L **do**
3   **while** *LossConvergence*$(\mathcal{F})$ not met **do**
4         $f_{ij} \leftarrow$*SelectNetwork*$(\mathcal{F})$ ▷ selects target network to be trained.
5         $p \leftarrow$*SelectPath*$(f_{ij}, k, \mathcal{P})$ ▷ selects a feasible consistency path for f_{ij} with maximum length k from $\mathcal{P}$.
6   optimize $\mathcal{L}^{\textit{perceptual}}_{ijp}$ ▷ trains f_{ij} using path constraint p in loss 8.
7   **end**
8 **end**

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[8]. 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[8]. 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 $\mathcal{F}_x$ and $\mathcal{F}_x$ 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 $\mathcal{F}_x$, rather than $\mathcal{F}_y$, one can first train $\mathcal{F}_y$ to convergence using Alg.1, then start the training of $\mathcal{F}_x$ with well trained and converged networks $\mathcal{F}_y$. 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 $\mathcal{Y}_k$, the consistency energy is defined to be the standardized average of pairwise inconsistencies:

$$\text{Energy}_{\mathcal{Y}_k}(x) \triangleq \frac{1}{|\mathcal{Y}|-1} \sum_{\mathcal{Y}_i \in \mathcal{Y}, i \neq k} \frac{|f_{\mathcal{Y}_i\mathcal{Y}_k} \circ f_{\mathcal{X}\mathcal{Y}_i}(x) - f_{\mathcal{X}\mathcal{Y}_k}(x)| - \mu_i}{\sigma_i}, \quad (9)$$

where μ_i and σ_i are the average and standard deviation of $|f_{\mathcal{Y}_i\mathcal{Y}_k} \circ f_{\mathcal{X}\mathcal{Y}_i}(x) - f_{\mathcal{X}\mathcal{Y}_k}(x)|$ 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

[8]See supplementary material for an experimental comparison.

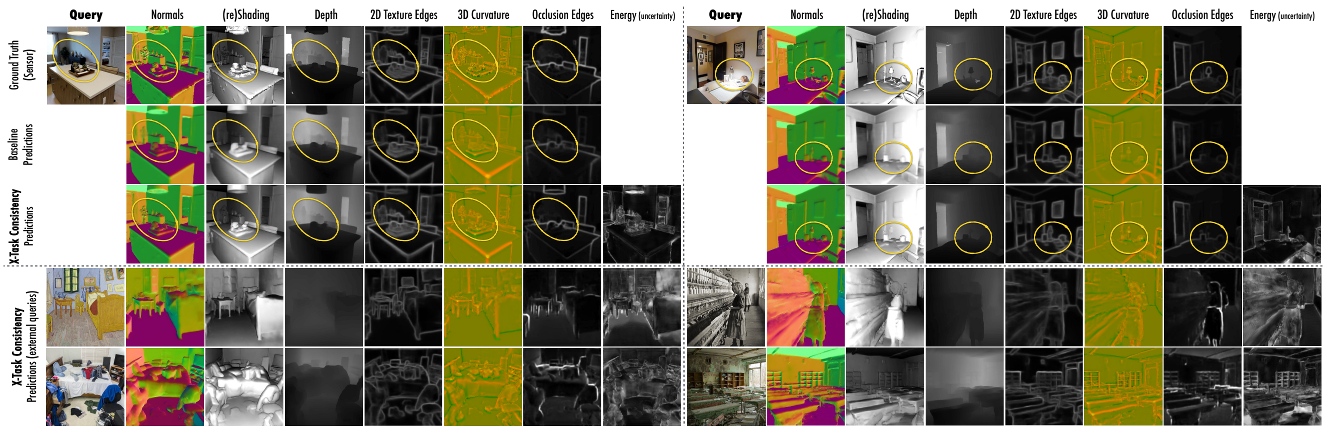


Figure 6: **Qualitative results of predicting multiple domains along with the pixel-wise Consistency Energy.** The **top** queries are from the Taskonomy dataset's test set. The results of networks trained with consistency are more accurate, especially in fine-grained regions (zoom into the yellow markers), and more correlated across different tasks. The **bottom** images are external queries (no ground truth available) demonstrating the generalization and robustness of consistency networks to external data. Comparing the energy against a prediction domain (e.g. normals) shows that energy often correlates with error. More examples are provided in the project page, and a live demo for user uploaded images is available at the demo page. **External Queries**: Bedroom in Arles, Van Gogh (1888); Cotton Mill Girl, Lewis Hine (1908); Chernobyl Pripyat Abandoned School (c. 2009). [best seen on screen]

variance in predictions in the lower row of Fig. 3 – the higher the variance, the higher the inconsistency, and the higher the energy. The consistency energy is an intrinsic quantity of the system and needs no ground truth or supervision.

In Sec. 5.3, we show this quantity turns out to be quite informative as it can indicate the reliability of predictions (useful as a confidence/uncertainty metric) or a shift in the input domain (useful for domain adaptation). This is based on the fact that if the query is from the same data distribution as the training and is unchallenging, all inference paths of a system trained with consistency path constraints work well and yield similar results (as they were trained to); whereas under a distribution shift or for a challenging query, different paths break in different ways resulting in dissimilar predictions. In other words, usually *correct* predictions are *consistent* while *mistakes* are *inconsistent*. (Plots 9(b), 9(c), 9(d).)

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 (zbuffer), 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 $\mathcal{F}_\mathcal{X}$ and $\mathcal{F}_\mathcal{Y}$ 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 3×10^{-5}, weight decay 2×10^{-6}, and batch size 32. Input and output images were linearly scaled to the range $[0, 1]$ and resized down to 256×256. We used ℓ_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 $\mathcal{Y}_2$=$\mathcal{X}$.

Baseline perceptual loss network uses frozen random (Gaussian weight) networks as $\mathcal{F}_\mathcal{Y}$, 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].

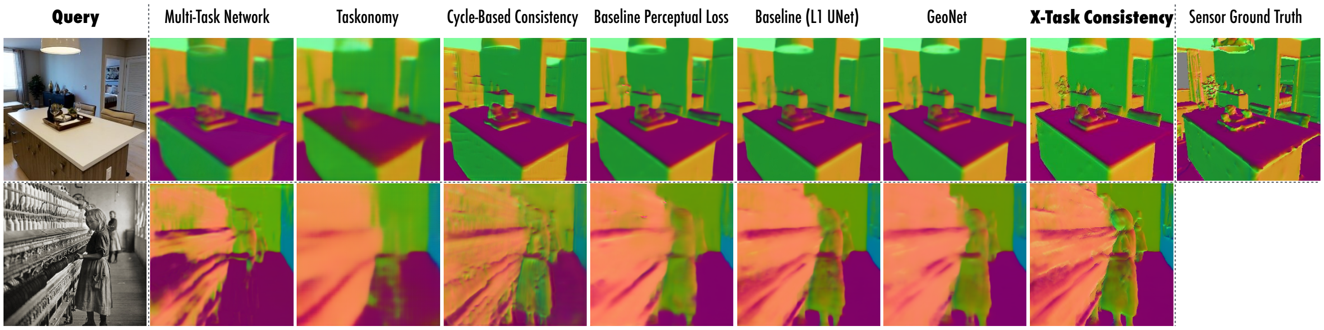


Figure 7: **Learning with cross-task consistency vs various baselines** compared over surface normals. Queries are from Taskonomy dataset (**top**) or external data (**bottom**). More examples are provided in project page, and a live demo for user uploaded images is available at demo page. [best seen on screen]

Replica Dataset:

Method	Normals: Perceptual Err. Depth	Normals: Perceptual Err. reShade	Normals: Direct ℓ_1 Err.	Depth: Perceptual Err. Norm.	Depth: Perceptual Err. reShade	Depth: Direct ℓ_1 Err.	reShading: Perceptual Err. Norm.	reShading: Perceptual Err. Depth	reShading: Direct ℓ_1 Err.
Blind Guess	4.75	33.31	16.02	22.23	19.94	4.81	15.74	5.14	16.45
Taskonomy Networks	3.73	11.07	6.55	18.06	15.39	3.72	8.70	3.85	11.43
Multi-Task	5.58	22.11	6.03	15.30	16.14	2.44	7.24	3.36	10.32
GeoNet (original)	6.23	19.34	7.48	13.88	14.03	4.01	×	×	×
Cycle Consistency	5.65	22.39	7.13						
Baseline Perceptual Loss	4.88	15.34	4.99						
Pix2Pix	4.52	19.03	7.70						
Baseline UNet (ℓ_1)	4.69	13.15	4.96	10.47	12.99	1.99	6.90	2.74	**9.55**
GeoNet (updated)	4.62	12.79	**4.70**	10.47	12.75	1.83	×	×	×
X-Task Consistency	**2.07**	**9.99**	**4.80**	**7.01**	**11.21**	**1.63**	**5.50**	**1.96**	**9.22**
0.25% Data: Baseline	5.65	21.76	7.61						
0.25% Data: Consistency	**2.41**	**12.26**	**7.28**						

Taskonomy Dataset (Normals, Depth):

Method	Normals: Perceptual Err. Depth	Normals: Perceptual Err. reShade	Normals: Perceptual Err. Curv.	Normals: Perceptual Err. Edge(2D)	Normals: Direct ℓ_1 Err.	Depth: Perceptual Err. Norm.	Depth: Perceptual Err. reShade	Depth: Perceptual Err. Curv.	Depth: Perceptual Err. Edge(2D)	Depth: Direct ℓ_1 Err.
Blind Guess	7.39	38.11	3.91	12.05	17.77	22.37	27.27	7.96	12.77	7.07
Taskonomy Networks	7.19	22.68	3.68	10.70	7.54	18.82	20.83	6.65	14.10	4.55
Multi-Task	8.78	27.32	3.65	10.16	7.07	17.18	19.55	7.54	13.67	2.81
GeoNet (original)	7.71	27.35	**3.32**	**9.09**	9.58	15.44	18.73	4.03	10.78	4.07
Cycle Consistency	8.81	30.33	3.84	10.26	8.68					
Baseline Perceptual Loss	8.59	23.98	3.41	10.01	6.17					
Pix2Pix	8.12	26.23	3.83	10.33	9.40					
Baseline UNet (ℓ_1)	8.17	20.94	3.41	9.98	**5.95**	13.62	15.68	7.31	12.61	**2.27**
GeoNet (updated)	8.18	20.84	3.40	9.99	**5.91**	13.77	15.76	7.52	12.67	**2.26**
X-Task Consistency	**4.32**	**12.15**	**3.29**	9.50	6.08	**9.46**	**12.66**	**3.61**	**9.82**	**2.29**
0.25% Data: Baseline	8.86	26.91	3.78	10.31	**8.17**					
0.25% Data: Consistency	**5.07**	**15.96**	**3.74**	**9.93**	9.19					

Taskonomy Dataset (reShading, Semantic Segm.):

Method	reShading: Perceptual Err. Norm.	reShading: Perceptual Err. Depth	reShading: Perceptual Err. Curv.	reShading: Perceptual Err. Edge(2D)	reShading: Direct ℓ_1 Err.	Semantic Segm.: Direct X-Entropy (↓)
Blind Guess	19.96	7.14	3.53	12.62	24.85	
Taskonomy Networks	11.72	4.69	3.54	11.19	16.58	
Multi-Task	9.19	3.54	3.56	10.75	11.61	
GeoNet (original)	×	×	×	×	×	×
Cycle Consistency						
Baseline Perceptual Loss						
Pix2Pix						
Baseline UNet (ℓ_1)	9.58	3.38	3.78	10.85	**10.45**	0.246
GeoNet (updated)	×	×	×	×	×	×
X-Task Consistency	**7.13**	**2.51**	**3.28**	**9.38**	**10.52**	**0.237**
0.25% Data: Baseline						
0.25% Data: Consistency						

Table 1: **Quantitative Evaluation of Cross-Task Consistent Learning vs Baselines.** Results are reported on Replica and Taskonomy Datasets for four prediction tasks (normals, 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, ℓ 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 (visuls 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.

5.1. 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. ℓ_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.[9] Each perceptual error provides a different angle, and the optimal results would have a low error for *all* metrics.

Tab. 1 also includes evaluation of the networks when trained with little data (0.25% subset of Taskonomy dataset), which shows the consistency constraints are useful under low-data regime as well.

We adopted normals as the canonical task for more extensive evaluations, due to its practical value and abundance of baselines. The conclusions remained the same regardless.

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.

[9]For example, evaluation of normals via the *normal→curvature* metric is akin to paying more attention to where normals change, hence reducing the domination of flat regions, such as walls, in the numbers.

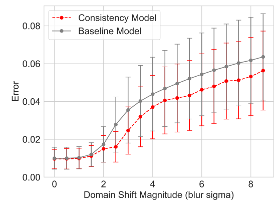
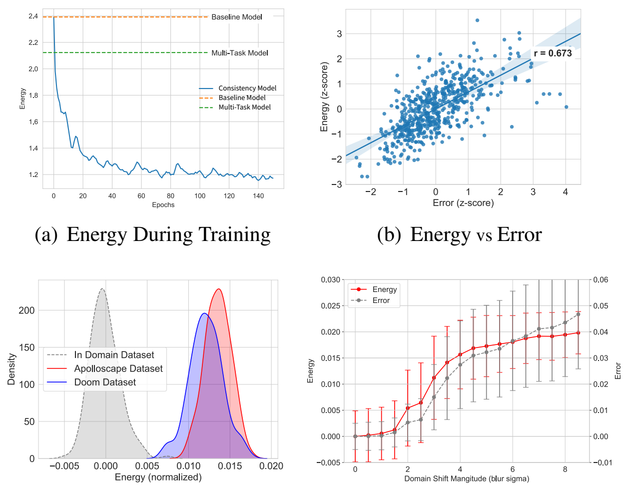


Figure 8: **Error with Increasing (Smooth) Domain Shift.** The network trained with consistency is more robust to the shift.

Novel Domain	Error (Post-Adaption) # images	Consistency	Baseline	Error (Pre-Adaptation) Consistency	Baseline
Gaussian blur (Taskonomy)	128	17.4 (+14.7%)	20.4	46.2 (+12.8%)	53.0
	16	22.3 (+8.6%)	24.4		
CocoDoom	128	18.5 (+19.2%)	22.9	54.3 (+15.8%)	64.5
	16	27.1 (+24.5%)	35.9		
ApolloScape	8	40.5 (+11.9%)	46.0	55.8 (+5.5%)	59.1

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)



(a) Energy During Training (b) Energy vs Error

(c)Energy vs Discrete Domain Shift (d) Energy vs Continuous Domain 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.

References

[1] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *CoRR*, abs/1812.10464, 2018. 2

[2] Richard W. Brislin. Back-translation for cross-cultural research. *Journal of Cross-Cultural Psychology*, 1(3):185–216, 1970. 2

[3] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018. 8

[4] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 2

[5] Luca Cosmo, Emanuele Rodola, Andrea Albarelli, Facundo Mémoli, and Daniel Cremers. Consistent partial matching of shape collections via sparse modeling. In *Computer Graphics Forum*, volume 36, pages 209–221. Wiley Online Library, 2017. 2

[6] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. *CoRR*, abs/1904.07846, 2019. 2

[7] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. *CoRR*, abs/1808.09381, 2018. 2

[8] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016. 2

[9] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017. 2

[10] V. Guillemin and A. Pollack. *Differential Topology*. Mathematics Series. Prentice-Hall, 1974. 2

[11] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017. 8

[12] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340. ACM, 2001. 2

[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

[14] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *CoRR*, abs/1711.03213, 2017. 2

[15] Qi-Xing Huang and Leonidas Guibas. Consistent shape maps via semidefinite programming. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*, SGP '13, pages 177–186, Aire-la-Ville, Switzerland, Switzerland, 2013. Eurographics Association. 2

[16] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. *CoRR*, abs/1803.06184, 2018. 6

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. 2, 6

[18] Jason Jo and Yoshua Bengio. Measuring the tendency of cnns to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*, 2017. 1, 8

[19] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016. 4

[20] Mykel J. Kochenderfer, Christopher Amato, Girish Chowdhary, Jonathan P. How, Hayley J. Davison Reynolds, Jason R. Thornton, Pedro A. Torres-Carrasquillo, N. Kemal Üre, and John Vian. *Decision Making Under Uncertainty: Theory and Application*. The MIT Press, 1st edition, 2015. 8

[21] Iasonas Kokkinos. Ubernet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. *CoRR*, abs/1609.02132, 2016. 2, 6

[22] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. 2

[23] Uday Kusupati, Shuo Cheng, Rui Chen, and Hao Su. Normal assisted stereo depth estimation. *arXiv preprint arXiv:1911.10444*, 2019. 2

[24] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017. 2

[25] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291, 2019. 2

[26] Yann LeCun, Sumit Chopra, and Raia Hadsell. A tutorial on energy-based learning. 2006. 5

[27] Aravindh Mahendran, Hakan Bilen, João F. Henriques, and Andrea Vedaldi. Researchdoom and cocodoom: Learning computer vision with games. *CoRR*, abs/1610.02431, 2016. 6

[28] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 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

[30] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph.*, 31(4):30:1–30:11, July 2012. 2

[31] F Proctor, Marek Franaszek, and J Michaloski. Tolerances and uncertainty in robotic systems. In *ASME 2017 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers Digital Collection, 2017. 8

[32] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018. 2, 7

[33] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019. 6

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 6

[35] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017. 8

[36] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014. 2

[37] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. *Indoor Segmentation and Support Inference from RGBD Images*, pages 746–760. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 6

[38] Trevor Standley, Amir R. Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which Tasks Should Be Learned Together in Multi-task Learning? *arXiv e-prints*, page arXiv:1905.07553, May 2019. 2

[39] James Stewart. *Essential calculus: Early transcendentals*. Cengage Learning, 2012. 3

[40] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 6

[41] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. *CoRR*, abs/1903.07593, 2019. 2

[42] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 6

[43] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018. 2

[44] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5684–5693, 2019. 2

[45] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018. 2, 6, 7

[46] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5287–5295, 2017. 2

[47] Yu Zhang and Qiang Yang. A survey on multi-task learning. *CoRR*, abs/1707.08114, 2017. 2

[48] Zaiwei Zhang, Zhenxiao Liang, Lemeng Wu, Xiaowei Zhou, and Qixing Huang. Path-invariant map networks. *CoRR*, abs/1812.11647, 2018. 2

[49] Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qixing Huang, and Alexei A. Efros. Learning dense correspondence via 3d-guided cycle consistency. *CoRR*, abs/1604.05383, 2016. 2

[50] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017. 2, 6, 8

[51] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 36–53, 2018. 2