# Multi-Task Label Discovery via Hierarchical Task Tokens for Partially Annotated Dense Predictions

Jingdong Zhang
Texas A&M University
College Station, Texas, USA
jdzhang@tamu.edu

Hanrong Ye
Hong Kong University of Science and Technology
Hong Kong, China
hanrong.ye@connect.ust.hk

Xin Li
Texas A&M University
College Station, Texas, USA
xinli@tamu.edu

Wenping Wang*
Texas A&M University
College Station, Texas, USA
wenping@tamu.edu

Dan Xu
Hong Kong University of Science and Technology
Hong Kong, China
danxu@cse.ust.hk



## Abstract

In recent years, simultaneous learning of multiple dense prediction tasks with partially annotated label data has emerged as an important research area. Previous works primarily focus on leveraging cross-task relations or conducting adversarial training for extra regularization, which achieve promising performance improvements, while still suffering from the lack of direct pixel-wise supervision and extra training of heavy mapping networks. To effectively tackle this challenge, we propose a novel approach to optimize a set of compact learnable hierarchical task tokens, including global and fine-grained ones, to discover consistent pixel-wise supervision signals in both feature and prediction levels. Specifically, the global task tokens are designed for effective cross-task feature interactions in a global context. Then, a group of fine-grained task-specific spatial tokens for each task is learned from the corresponding global task tokens. It is embedded to have dense interactions with each task-specific feature map. The learned global and local fine-grained task tokens are further used to discover pseudo task-specific dense labels at different levels of granularity, and they can be utilized to directly supervise the learning of the multi-task dense prediction framework. Extensive experimental results on challenging NYUD-v2, Cityscapes, and PASCAL Context datasets demonstrate significant improvements over existing state-of-the-art methods for partially annotated multi-task dense prediction.

## CCS Concepts

• **Computing methodologies** → **Computer vision tasks**.

*Corresponding author.


*MM '25, Dublin, Ireland*

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755727

## Keywords

Multi-task Partially Annotated Dense Prediction, Multi-task Learning, Label Discovery

**ACM Reference Format:**
Jingdong Zhang, Hanrong Ye, Xin Li, Wenping Wang, and Dan Xu. 2025. Multi-Task Label Discovery via Hierarchical Task Tokens for Partially Annotated Dense Predictions. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3746027.3755727

## 1 Introduction

With the rapid development of supervised learning with deep neural networks, various pixel-wise dense prediction tasks with highly complementary properties such as semantic segmentation and depth estimation have achieved great success in multi-task learning (MTL) in recent years [40, 57, 66, 71, 80]. Researchers pursue the learning of them simultaneously in a unified framework, which can effectively model cross-task correlations and achieve superior results in terms of model training costs and performances.

However, in real-world scenarios, obtaining pixel-level annotations is prohibitively expensive, especially when dealing with a set of distinct dense prediction tasks. Each image has to be annotated with pixel labels for all the tasks. Thus, existing works have delved into the problem of multi-task learning with only partially annotated dense labels [28, 36, 41, 62, 74, 75, 77]. Specifically, as illustrated in Fig. 1 (a), given an input image, for $T$ dense prediction tasks, the task labels are provided partially, i.e. for at least one task and at most $T-1$ tasks. Learning a multi-task model under this setting is particularly challenging since every input image lacks some of the task supervision signals, and the performance typically drops significantly if compared to the same model trained with full task label supervisions [28].

Previous works have been focusing on excavating cross-task relations by training heavy extra mapping networks [28, 41], however, simply applying regularization in compact latent spaces fails to address the lack of dense pixel supervision and limits the performance. On the contrary, directly discovering pseudo task labels in prediction spaces can alleviate this problem to a certain extent, while still suffering from the following two severe limitations: (i) Simply discovering task labels in the prediction spaces separately

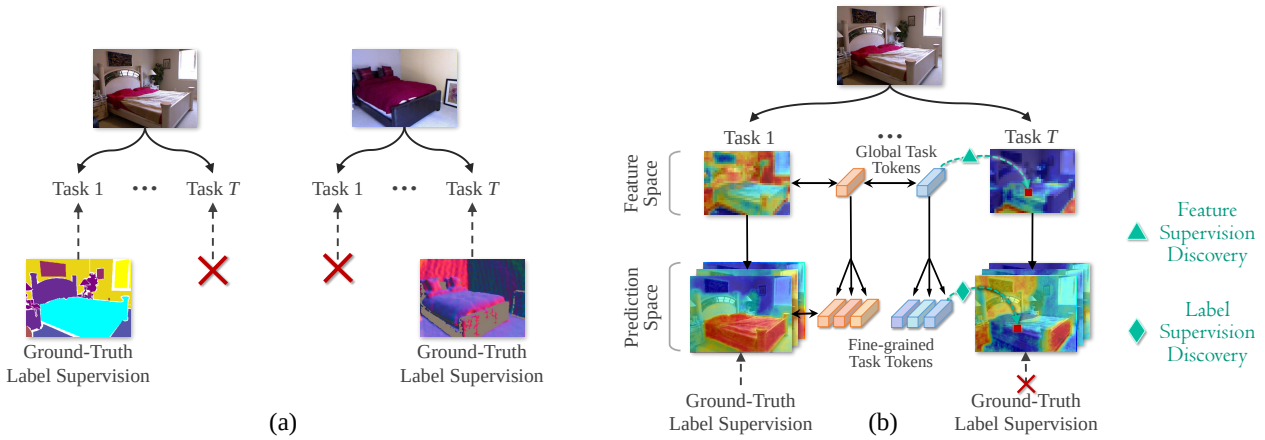


**Figure 1: (a) Illustration of partially annotated multi-task dense prediction setting. Each input image only has partial task labels from all the tasks. (b) Illustration of the learning of Hierarchical Task Tokens, including global task tokens and fine-grained task tokens, by conducting feature-token interactions in feature and prediction spaces separately. The well-learned hierarchical task tokens can achieve both feature supervision discovery and task label discovery.**

ignores the highly relevant task relations, thus leading to trivial performance under multitask scenarios. (ii) Solely discovering labels in prediction spaces cannot take advantage of the abundant task representations in feature space. For each specific task, the distributions of the task features and predictions should be consistent. Since the encoder parameters are shared among tasks, thus the produced rich task-generic features are beneficial for building cross-task relations and discovering supervision signals effectively. Therefore, it is critically important to involve hierarchical cross-task representations from feature space to prediction space to consistently boost the task label discovery process.

To effectively tackle the aforementioned challenges, we propose a novel approach that performs task label discovery from both the feature and prediction spaces via an effective design of learnable Hierarchical Task Tokens (HiTTs). HiTTs are sets of compact parameters that are learned in a hierarchical manner to model global inter-task relationships and local fine-grained intra-task relationships, which allows for discovering pixel-wise task pseudo labels straightforwardly in both feature and prediction spaces consistently. More specifically, as depicted in Fig. 1 (b), we apply HiTTs during the multi-task decoding stage and jointly optimize them with the multi-task learning network. The HiTTs consist of two hierarchies. The first hierarchy is a set of global task tokens. The global task tokens are randomly initialized and can perform cross-task feature-token interactions with different task feature maps based on self-attention. These learned task tokens can be used to discover feature-level pseudo supervision by selecting highly activated pixel features correlated to each task. The second hierarchy is the fine-grained task tokens. These tokens are directly derived from the global task tokens with learnable projection layers to inherent beneficial task representations. They are subsequently utilized to perform interactions within each task-specific feature map at a finer granularity. As the fine-grained task tokens can learn pixel-to-pixel correlation with each task feature map, they thus benefit the discovery of dense spatial labels for each task. Compared with naive pseudo labeling process [26], the HiTTs can bridge effective information from supervised tasks so as to encourage highly confident predictions on unsupervised tasks. We learn both hierarchies of tokens simultaneously in an end-to-end manner incorporating the multi-task baseline network, and exploit both levels of supervision signals discovered from the two hierarchies, for optimizing multi-task dense predictions on partially annotated datasets.

In summary, the contribution of this work is three-fold:

- Instead of discovering cross-task regularization by extra heavy mapping networks, we propose to utilize cross-task relations for high-quality task label discovery, which serves as pixel-level dense pseudo supervision under the multi-task partially supervised setting.
- We propose a novel design of Hierarchical Task Tokens (HiTTs), which can learn hierarchical multi-task representations for high-quality pseudo label discovery consistently in both the feature and the prediction levels.
- Our proposed method significantly outperforms existing state-of-the-art competitors on multi-task partially annotated benchmarks, including PASCAL-Context, NYUD-v2 and Cityscapes, and demonstrates clear effectiveness on challenging dense prediction tasks with limited annotations, including segmentation, depth estimation, normal estimation and edge detection, etc. Code is released at https://github.com/Evergreen0929/EEMTL.

## 2 Related Work

**Multi-task Dense Prediction.** Dense prediction tasks aim to produce pixel-wise predictions for each image. Common tasks including semantic segmentation, depth estimation, and surface normal estimation exhibit high cross-task correlations. For instance, depth discontinuity is usually aligned with semantic boundaries [57], and surface normal distributions are aligned with spatial derivatives of the depth maps [35]. Thus, a number of works have been focusing on multi-task dense predictions [6, 7, 18, 33, 34, 40, 53, 55, 57, 66, 69–73, 78–81]. They leverage parameters sharing to conduct cross-task interactions by effective attention mechanisms for task feature

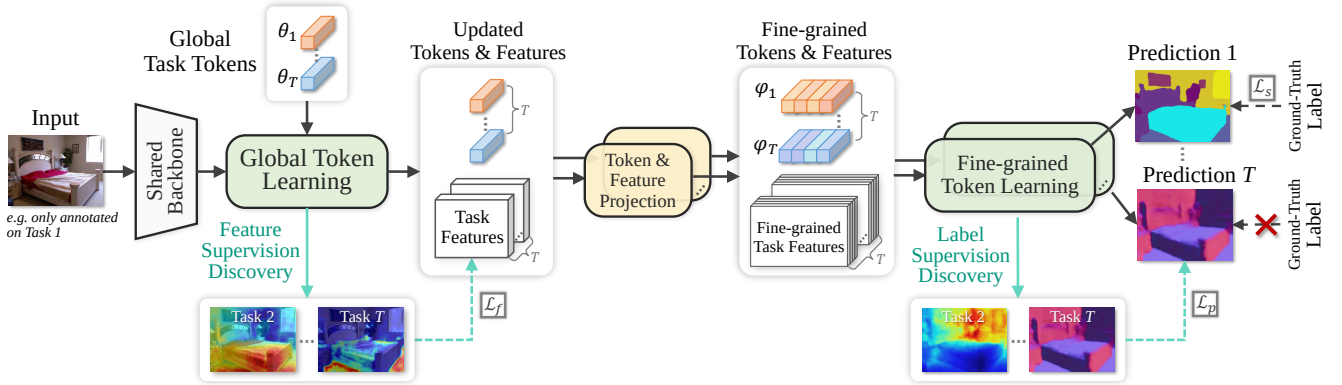


**Figure 2: Illustration of our method. HiTTs consist of both global and fine-grained task tokens which learn discriminative task representations by conducting feature-token interactions with attentions in corresponding multi-task decoding stages. The global task tokens $\theta_i$ discover feature-level pseudo supervision $\mathcal{L}_f$, while the fine-grained task tokens $\varphi_i$ inherit the knowledge from global task tokens and directly discover pixel labels for supervision $\mathcal{L}_p$. The supervision from ground-truth label is denoted as $\mathcal{L}_s$.**

selection [33], and multi-modal distillation [66], multi-scale cross-task interactions [57], global pixel and task interactions [71] and multi-task bridge features [79]. However, these works focus on fully-supervised settings. In contrast, our work addresses the challenge of insufficient supervision signals in each task.

**Semi-supervised learning.** Obtaining pseudo labels for semi-supervised learning is a popular research direction, with several deep learning works published on the topic [23, 26, 29, 47, 49, 54, 65, 84, 85]. Among them, [26] aims at picking up the class which has the maximum predicted confidence. The graph-based label propagation method [23] is also used to infer pseudo-labels for unlabeled data. [47] provides a confidence level for each unlabeled sample to reduce influences from outliers and uncertain samples, and uses MMF regularization at feature levels to make images with the same label close to each other in the feature space. [65] uses accurate pseudo labels produced by the teacher model on clean unlabeled data to train the student model with noise injected. For semi-supervised dense prediction tasks [21, 37, 43, 44, 67, 68], such as semantic segmentation, several works focus on assigning pixel-wise pseudo annotations from high-confidence predictions [29, 84, 85]. However, these works target single-task learning setups. Despite pseudo labeling, discovering consistency for regularization is also a popular direction for unlabeled data [28, 36, 52, 77]. [28, 36, 77] focus on building cross-task consistency, while [52] uses image-level feature similarities to find important samples for semi-supervised learning. Differently, our work targets *pixel-level* task supervision discovery by hierarchical task tokens containing multi-level multi-task representations for partially annotated dense predictions.

**Multi-task Partially Supervised Learning.** As discussed in the introduction, obtaining pixel-level annotations for every task on images is prohibitively expensive. Therefore, some recent works focus on partially annotated settings for multi-task learning [22, 28, 32, 35, 36, 41, 62, 74, 75, 77]. Since directly recovering labels from other tasks is an ill-posed problem [28], enforcing consistency among tasks is usually adopted. For instance, constructing a common feature space to align predictions and impose regularization [28], and leveraging intrinsic connections of different task pairs between predictions of different tasks on unlabeled data in a mediator dataset, when jointly learning multiple models [35]. Adversarial training is also adopted to align the distributions between labeled and unlabeled data by discriminators [62], and multi-task denoising diffusion is adopted [74] to address the issue of noise in initial prediction maps. To the best of our knowledge, our hierarchical task tokens for both pseudo feature supervision and task label discovery are a novel exploration of the problem, and show a clear difference from existing works.

## 3 Proposed Method

Our proposed approach for learning Hierarchical Task Tokens (HiTTs) primarily comprises two stages, *i.e.*, the Global Token Learning and the Fine-grained Token Learning. The overall structure of HiTTs is depicted in Fig. 2. Firstly, in the Global Token Learning stage, the global task tokens produce task features and then learn rich task-level representations by conducting inter- and intra-task attention with all task features. The global tokens are utilized to exploit rich representations in feature space and discover feature-level pseudo supervision. Subsequently, in the fine-grained stage, we project each task feature into fine-grained feature space by simple convolution layers, and derive the fine-grained tokens from the global tokens by Multi-layer Perceptrons (MLPs), to inherit well-learned global task representations and therefore achieve consistent pseudo label discovery. To perform a uniform confidence-based pseudo label discovery for different types of dense prediction tasks, we follow [4] to conduct discrete quantization of regression task annotations (e.g. depth estimation and normal estimation), and treat all tasks as pixel-wise classification.

### 3.1 Global Task Token Learning

In the global task token learning stage, we target learning global tokens representing the distributions of each task, which are further used for pseudo feature supervision discovery. The learning process is mainly achieved by inter- and intra-task attention among tokens and features, in order to exploit beneficial multi-task representations for token learning and feature supervision discovery.

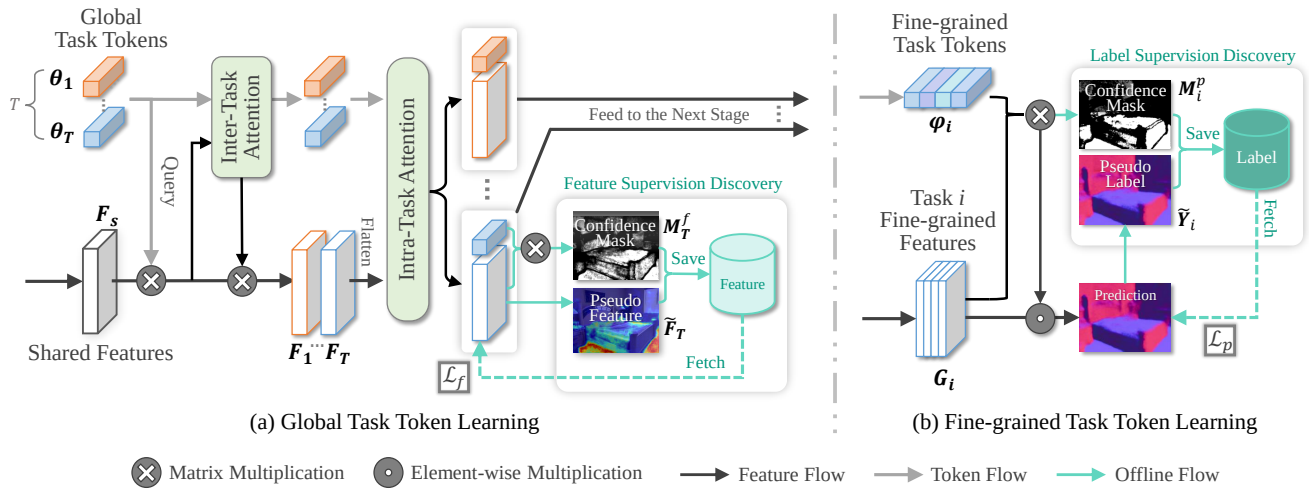


**Figure 3: (a) Global Task Token Learning. It mainly contains two stages: Inter- and Intra-Task Attention for thoroughly feature-token cross-task interaction to obtain robust and representative global task tokens, and excavating pseudo feature supervision based on the learned task tokens. (b) Fine-grained Task Token Learning. The projected fine-grained tokens and feature maps (projection is shown in Fig. 4) can be used for discovering high-quality pseudo labels in prediction spaces.**

Given an RGB input $\boldsymbol{X} \in \mathbb{R}^{3\times H\times W}$, a multi-task dense prediction framework firstly produces a task-generic representation $\boldsymbol{F_s} \in \mathbb{R}^{C\times h\times w}$ through a shared encoder. Considering we have $T$ tasks, and we target decoding task features $\{\boldsymbol{F_1}, \boldsymbol{F_2}, \cdots, \boldsymbol{F_T}\}$ from $\boldsymbol{F_s}$ as well as learning representative global task tokens $\{\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \cdots, \boldsymbol{\theta_T}\}$, $\boldsymbol{\theta_i} \in \mathbb{R}^C$ for each task. They are randomly initialized learnable vectors serving as additional input tokens for the decoder.

As shown in Fig. 3 (a), the proposed global task token learning process is mainly composed of two stages: i) Inter- and Intra-Task Attention, which aim to thoroughly model feature-token cross-task relations to obtain robust and representative global task tokens. ii) Feature Supervision Discovery, which excavates pseudo feature supervision with confidence maps provided by global task tokens for unsupervised task features.

Firstly, we use each global task token to query the shared feature to obtain each task feature $\boldsymbol{F_i}$ accordingly. Then, to conduct *inter-task attention*, all global task tokens are used to calculate all-task affinities $\boldsymbol{A} \in \mathbb{R}^{T\times T}$ to represent the global task relations. After the $\boldsymbol{A}$ is calculated, it is used to conduct affine combinations of task features and global task tokens respectively. Since for each task feature $\boldsymbol{F_i}$, if it is not directly supervised by labels, the feature will be less representative and contain more noise. Thus, conducting affine combinations among all tasks ensures that beneficial discriminative representations from other supervised task features can fertilize the unsupervised ones. Afterward, the updated tokens and features containing cross-task information are fed to the *intra-task attention* module, where they are rearranged and grouped in each task, and self-attention is applied to each group of task token and feature. The global task tokens will further learn more specific and discriminative task representations during this process, and representative task tokens will in turn enhance the feature quality as well.

Followingly, we discover pseudo feature supervision with the aid of well-learned global task tokens, and this process will be discussed in Sec. 3.3. In addition, for multi-scale backbone features, directly fusing them ignores the various granularity of task representations maintained at different scales. Thus, for multi-scale image backbone, we further propose Multi-scale Global Task Token Learning in order to learn comprehensive multi-scale task relations. The proposed method involves *inter-task attention* separately at each scale, and then the multi-scale features and tokens are fused before *intra-task attention*. In this way, the global task tokens gain richer cross-task relations at different scales and are able to maintain stronger representations. The multi-scale global task token learning, inter- and intra-task attention will be illustrated in detail in the supplementary material.

## 3.2 Fine-grained Task Token Learning

After the global task tokens are learned, we further propose to conduct feature-token interaction at a finer spatial granularity, which takes advantage of various representations in global task tokens and boosts the task label discovery process.

Firstly, as shown in Fig. 4, we jointly project each updated task token $\boldsymbol{\theta_i} \in \mathbb{R}^C$ and feature $\boldsymbol{F_i} \in \mathbb{R}^{C\times h\times w}$ into the prediction space with finer granularity. For features, this can be easily achieved by applying a linear convolution layer, and we denote the fine-grained task features as $\boldsymbol{G_i} \in \mathbb{R}^{C_p\times H\times W}$, where $C_p$ indicates the prediction dimension. For tokens, we denote the projected fine-grained tokens as $\boldsymbol{\varphi_i} \in \mathbb{R}^{C_p\times C}$, we hope every $1\times C$ vector inside it can represent one category distribution over the spatial dimension. The simplest way is to project each $\boldsymbol{\theta_i}$ with a Multi-Layer Perceptron (MLP), which can be described as: $\boldsymbol{\varphi_i} = \mathrm{MLP_i}(\boldsymbol{\theta_i})^\top, i = 1, 2, \cdots, T$. However, since there is no direct supervision imposed to distinguish every fine-grain token during this process, the MLPs will tend to degenerate and perform linearly correlated outputs, which prevents the fine-grained tokens from learning discriminative task-specific representations. To alleviate this problem, we propose to use Orthogonal Embeddings (OE) to serve as priors and aid the learning process of fine-grained tokens.

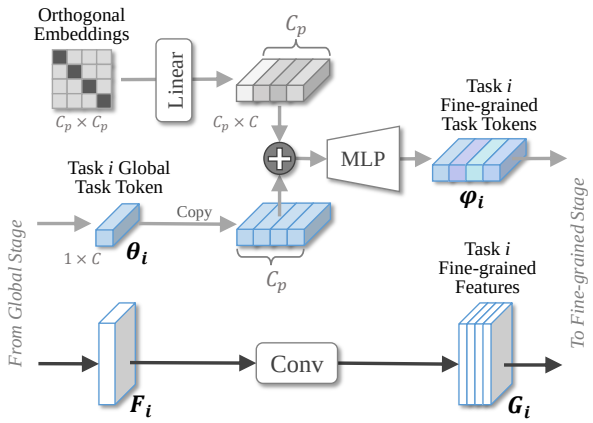


**Figure 4: Illustration of Token & Feature Projection and the Fine-grained Task Token Learning. Different tasks have the same structure, and we take one as an example. We project the fine-grained feature $G_i$ from the updated task feature $F_i$, and derive fine-grained task tokens $\varphi_i$ from the updated global task tokens $\theta_i$.**

In detail, the vectors in the fine-grained token should be far from each other to represent meaningful and distinguishable task category information, so we use a group of orthogonal basis in $\mathbb{R}^{C_p}$ to serve as the embedding for MLP input, denoting as $\boldsymbol{o} \in \mathbb{R}^{C_p \times C_p}$. These OE are projected into the feature space by linear projections, and then added with the global token before being fed into the MLP. Therefore, with these orthogonal priors, the MLPs can easily keep the distance between the vectors in $\boldsymbol{\varphi_i}$ far from each other in the feature space, which makes it able to learn information that distinguishes between task categories, as well as inherit the global representations from $\boldsymbol{\theta_i}$.

Subsequently, we exploit the fine-grained task tokens $\boldsymbol{\varphi_i}$ for the fine-grained token learning process. Normally, $\boldsymbol{G_i}$ is noisy and low-confident on unlabeled data due to the lack of supervision, which leads to inaccurate predictions. We propose to use fine-grained task tokens to encourage $\boldsymbol{G_i}$ to produce high-confident logits, and enhance the quality of pseudo labels:

$$\boldsymbol{G'_i} = \mathrm{Conv}_{\mathrm{i}}^{3\times3}\left(\boldsymbol{G_i} \odot \mathrm{Softplus}\left(\boldsymbol{\varphi_i} \times \boldsymbol{G_i}\right)\right), \tag{1}$$

where $\mathrm{Softplus}(x) = \log(1 + \exp(x))$. Since $\boldsymbol{\varphi_i}$ inherits global task representations from $\boldsymbol{\theta_i}$, which will perform more robustly on unlabeled data, and aid the production of distinguished logits score in the updated feature $\boldsymbol{G'_i}$, as well as high-confident final task predictions. Similar to the previous stage, we also conduct pseudo label discovery after the fine-grained tokens are learned, which will be discussed in detail in the next section.

## 3.3 Hierarchical Label Discovery and Multi-task Optimization

For the multi-task partially annotated setting, the training loss on labeled data can be described as:

$$\mathcal{L}_s = \frac{1}{T}\sum_{i=1}^{T}\left(\alpha_i \mathrm{L_i}\left(\hat{\boldsymbol{Y}}_i, \boldsymbol{Y_i}\right)\right), \tag{2}$$

where where $\mathrm{L_i}(\cdot)$ is the loss function for task $i$, and $\alpha_i = 1$ if task $i$ has ground-truth label, otherwise $\alpha_i = 0$. $\hat{\boldsymbol{Y}}_i$ and $\boldsymbol{Y_i}$ are task prediction and ground-truth.

We first train the multi-task model with HiTTs jointly with $\mathrm{L_i}$ only to achieve convergence on labeled data, then we utilize both hierarchies of tokens to discover feature-level and prediction-level pseudo supervision. As we mentioned in Sec. 3.1, before the updated tokens and features are fed into the next stage, we conduct the *feature supervision discovery* to excavate feature-level supervision signals for unlabeled tasks. As shown in Fig. 3 (a), we use the updated global task tokens to query each pixel feature on every task and produce confidence mask $\boldsymbol{M_i^f} = \mathrm{Sigmoid}(\boldsymbol{\theta_i}^\top \times \boldsymbol{F_i})$. Since $\boldsymbol{\theta_i}$ is globally learned on all task features, in $\boldsymbol{M_i^f}$, higher scores indicate that the pixel features have a higher response to task $i$, which should be further used to prove task supervision. Thus we use $\boldsymbol{M_i^f}$ to serve as a soft confidence mask for pixel-wise feature supervision loss:

$$\mathcal{L}_f = \frac{1}{T}\sum_{i=1}^{T}\left(\alpha_i \mathrm{L_{mse}}\left(\boldsymbol{F_i}, \tilde{\boldsymbol{F}}_i\right) + (1-\alpha_i)\boldsymbol{M_i^f} \odot \mathrm{L_{mse}}\left(\boldsymbol{F_i}, \tilde{\boldsymbol{F}}_i\right)\right), \tag{3}$$

where $\tilde{\boldsymbol{F}}_i$ represents the offline saved features which serve as pseudo supervision signals for unsupervised task features. The $\odot$ represents element-wise multiplication, $\mathrm{L_{mse}}$ is the mean squared error loss for feature distance measurement. For the feature loss on labeled task (first item in Eq 3), we regard all pixel features from $\tilde{\boldsymbol{F}}_i$ as valid since they are supervised by ground-truth label, while for unlabeled task (second item in Eq 3), we use $\boldsymbol{M_i^f}$ encourage high-confidence pixel features and depress low-confidence ones.

Afterward, we also conduct *pseudo label discovery* with the aid of fine-grained task tokens $\boldsymbol{\varphi_i}$ as mentioned in Sec. 3.2. We directly produce pseudo labels from $\boldsymbol{G'_i}$: $\tilde{\boldsymbol{Y}}_i = \mathrm{Argmax}\left(\mathrm{Softmax}\left(\boldsymbol{G'_i}\right)\right)$, along with binary masks to select high confidence pixel pseudo labels: $\boldsymbol{M_i^p} = \mathrm{Max}\left(\mathrm{Softmax}\left(\boldsymbol{G'_i}\right)\right) > \tau_i$, where $\tau_i$ is a threshold used to produce binary masks. The loss for pseudo label supervision can be written as:

$$\mathcal{L}_p = \frac{1}{T}\sum_{i=1}^{T}\left((1-\alpha_i)\boldsymbol{M_i^p} \odot \mathrm{L_i}\left(\hat{\boldsymbol{Y}}_i, \tilde{\boldsymbol{Y}}_i\right)\right), \tag{4}$$

Finally, we sum all of the losses in Eq 2, 3 and 4 to supervise all task features and predictions. The overall losses to optimize the model can be described as: $\mathcal{L} = \mathcal{L}_s + \mathcal{L}_p + \mathcal{L}_f$, each item is combined with the weight 1 to form the total loss.

# 4 Experiment

## 4.1 Experimental Setup

**PASCAL-Context.** PASCAL-Context [17] contains 4998 and 5105 images for training and testing respectively, which also have pixel-level annotations for semantic segmentation, human-parts segmentation and semantic edge detection. Additionally, we also consider surface normal estimation and saliency detection distilled by [38]. We use Adam optimizer with learning rate $2 \times 10^{-5}$, and weight decay $1 \times 10^{-6}$, and train for 100 epochs with batch size 6. We update the learning rate with polynomial strategy and $\gamma = 0.9$ for the power factor.

**NYUD-v2.** NYUD-v2 [48] contains 795 and 654 RGB-D indoor scene images for training and testing respectively. We use the 13-class semantic annotations which is defined in [13], the truth depth annotations recorded by Microsoft Kinect depth camera, and surface normal annotations which are produced in [16]. Following the setting in [28, 33], we use $288 \times 384$ image resolution to speed up training. We use Adam optimizer with a learning rate of $1 \times 10^{-4}$, and train all models for 400 epochs with batch size 8. We update the learning rate every 100 epoch with $\gamma = 0.5$ as the multiplying factor.

**Cityscapes.** Cityscapes [12] contains 2975 and 500 street-view images for training and testing respectively. We used the projected 7-class semantic annotations from [33], and disparity maps to serve as depth annotations. Following the setting in [28, 33], we use $128 \times 256$ image resolution to speed up training. The optimizer and learning rate scheduler are set as the same as NYUD-v2.

**Model Setting.** Following [28], we use SegNet [1] for NYUD-v2 and Cityscapes, ResNet-18 [20] for PASCAL-Context as the backbone of our single task learning (STL) baselines, and the multi-task baseline (MTL) is built from it, which consists of a shared backbone encoder and several task-specific decoding heads. For the learning of HiTTs, we follow [4, 27, 76], and perform a discrete quantization of the label space of continuous regression tasks such as Depth. and Normal. This discrete quantization does not contribute to multi-task learning performance as analyzed in [4], so we ensure a fair comparison with other works.

**Data Preparation.** We follow the setting of [74] to process PASCAL-Context, and the setting of [28, 33, 41] to process NYUDv2 and Cityscapes, and form two partially annotated settings [28]: **(i) one-label**: for each input image, it is only associated with one task annotation; **(ii) random-labels**: each image has at least one and at most $N - 1$ tasks with corresponding task annotations, in the set of $N$ tasks. Additionally, we provide two extra settings **full-labels** and **few-shot** in the supplementary material to further validate the effectiveness of our method.

**Training Pipeline.** We first train the multi-task model with HiTTs on all labeled task data. Then the weights of the network and tokens are fixed, and used to produce hierarchical supervision on both feature and prediction spaces. We produce the pseudo label in an offline manner according to [65], which is labeling on clean image without data augmentation, and training on augmented images and pseudo labels to enforce consistent predictions. In [65], this method is only applied to classification tasks while we extend the utilization to general dense prediction tasks. After the pseudo labels are saved, we use them along with the ground-truth labels to jointly train the multi-task model from scratch. Additionally, we also use the pseudo feature supervision produced by this pretrained multi-task model for feature regularization during the optimization process. Both hierarchies of the discovered supervision signals ensure that all task predictions will obtain pixel-wise supervision for multi-task optimization to gain better generalization ability on unlabeled data.

**Evaluation Metrics.** We use multiple metrics for each task to evaluate the performance. The metrics include: mIoU (mean intersection over union), AbS / AbR (absolute error / absolute-relative error), maxF (maximal F-measure), mErr (mean of angle error), odsF (optimal dataset scale F-measure). Additionally, to better evaluate the proposed method, we also use $\Delta_{MTL}$ from [56] to evaluate the overall improvement of the multi-task performances of all the tasks.

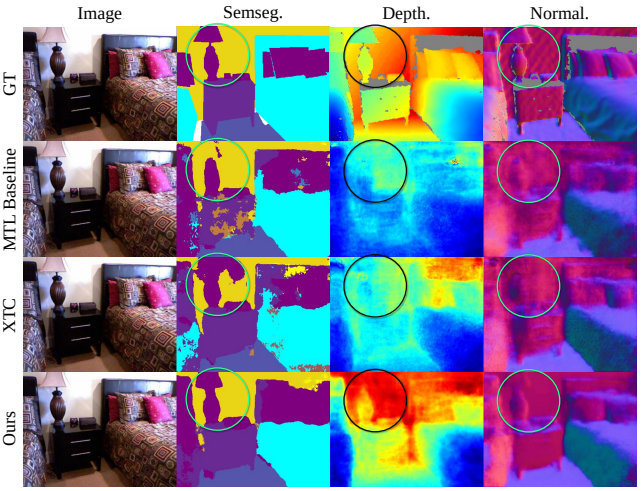


**Figure 5: Comparisons with SoTA works on NYUDv2. Ours shows both clear semantic boundaries and accurate geometry estimations, indicating the effectiveness of cross-task feature-token interactions.**

## 4.2 State-of-the-art Comparison

**Comparison on Pascal-Context.** For the comparison on Pascal-Context, we consider both *one-label* and *random-labels* settings. As shown in Table 1, our method achieves clear improvement over other methods on the majority of tasks under both settings. The greatest enhanced task is Semseg, which has +3.46 and +2.64 *mIoU* on the two settings compared with [74] (F). Overall, our method is +2.44% and +1.46% higher in terms of $\Delta_{MTL}$ compared with [74] (F). Moreover, our HiTTs are super compact compared with previous methods which either require heavy mapping networks [28, 41] or extra MTDNet for diffusion decoding [74], while ours achieve significantly better performance with $\sim$ 45% parameter amount and $\sim$ 70% GFlops compared with the best performing [74].

**Comparison on NYUD-v2.** We compare our method with [28, 33] on NYUD-v2 under both the one-label and random-labels settings, and the quantitative results are shown in Table 2. XTC [28] is the first work designed for partially annotated multi-task dense prediction. MTAN [28] is an attention-based MTL network designed for the fully supervised setting, and we train it with our setup. The quantitative results show that our method surpasses them by a large margin on all the metrics of the three tasks. More specifically, ours achieves +6.45% $\Delta_{MTL}$ and +7.41% $\Delta_{MTL}$ compared with [28] under the two partial-label settings, respectively. The qualitative comparison with the state-of-the-art method XTC [28] as shown in Fig. 5 can also confirm the superior performance of our method.

**Comparison on Cityscapes.** We also compare our results with [28, 33] on Cityscapes, under the *one-label* setting with both Semseg. and Depth. tasks. As shown in Table 3, our method achieves SOTA performance on both tasks, and significantly better performance on Depth (15.09% higher than [28]), resulting in an average gain of +8.76% in terms of $\Delta_{MTL}$. Additionally, it's worth mentioning that

**Table 1: Quantitative comparison on PASCAL-Context under the *one-label* and *random-labels* setting. (P) and (F) represent the Prediction Diffusion and Feature Diffusion modes for DiffusionMTL [74], and MTDNet is the Multi-Task Denoising Diffusion Networkt [74]. The Mapping Network is the extra encoder used for task mappings in [28, 41]. "*" denotes the re-implemented results from [74]. Our method performance outperforms previous methods while using significantly fewer model parameters.**

| # labels | Method | MTDNet | Mapping Network | #Params | FLOPS | Semseg mIoU ↑ | Parsing mIoU ↑ | Saliency maxF ↑ | Normal mErr ↓ | Boundary odsF ↑ | MTL Perf $\Delta_{MTL}(\%)$ ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| One-Label | STL | ✗ | ✗ | 219M | 817G | 50.34 | 59.05 | 77.43 | 16.59 | 64.40 | - |
| | MTL *baseline* | ✗ | ✗ | 157M | 608G | 49.71 | 56.00 | 74.50 | 16.85 | 62.80 | -2.85 |
| | SS [28] | ✗ | ✗ | - | - | 45.00 | 54.00 | 61.70 | 16.90 | 62.40 | - |
| | XTC [28] | ✗ | ✓ | - | - | 49.50 | 55.80 | 61.70 | 17.00 | 65.10 | - |
| | XTC* [28] | ✗ | ✓ | 173M | 608G | 55.08 | 56.72 | 77.06 | 16.93 | 63.70 | 0.37 |
| | JTR* [41] | ✗ | ✓ | 173M | 608G | 50.29 | 54.78 | 78.35 | 17.97 | 63.66 | -3.12 |
| | DiffusionMTL (P) [74] | ✓ | ✗ | 133M | 628G | 59.43 | 56.79 | 77.57 | 16.20 | 64.00 | 3.23 |
| | DiffusionMTL (F) [74] | ✓ | ✗ | 133M | 676G | 57.78 | **58.98** | 77.82 | 16.11 | 64.50 | 3.65 |
| | Ours | ✗ | ✗ | 62M | 493G | **61.24** | 57.52 | **78.35** | **15.75** | **67.70** | **6.09** |
| Random-Labels | STL | ✗ | ✗ | 219M | 817G | 51.51 | 57.90 | 80.30 | 15.24 | 67.80 | - |
| | MTL *baseline* | ✗ | ✗ | 157M | 608G | 62.23 | 55.88 | 78.67 | 15.47 | 66.70 | 2.44 |
| | SS [28] | ✗ | ✗ | - | - | 59.00 | 55.80 | 64.00 | 15.90 | 66.90 | - |
| | XTC [28] | ✗ | ✓ | - | - | 59.00 | 55.60 | 64.00 | 15.90 | 67.80 | - |
| | XTC* [28] | ✗ | ✓ | 173M | 608G | 62.44 | 55.81 | 78.56 | 15.45 | 66.80 | 2.52 |
| | JTR* [41] | ✗ | ✓ | 173M | 608G | 57.21 | 53.18 | 79.98 | 16.48 | 66.20 | -1.60 |
| | DiffusionMTL (P) [74] | ✓ | ✗ | 133M | 628G | 63.68 | 55.84 | 79.87 | 15.38 | 66.80 | 3.44 |
| | DiffusionMTL (F) [74] | ✓ | ✗ | 133M | 676G | 62.55 | **56.84** | 80.44 | 14.85 | 67.10 | 4.27 |
| | Ours | ✗ | ✗ | 62M | 493G | **65.19** | 56.35 | **81.70** | **14.80** | **67.90** | **5.73** |

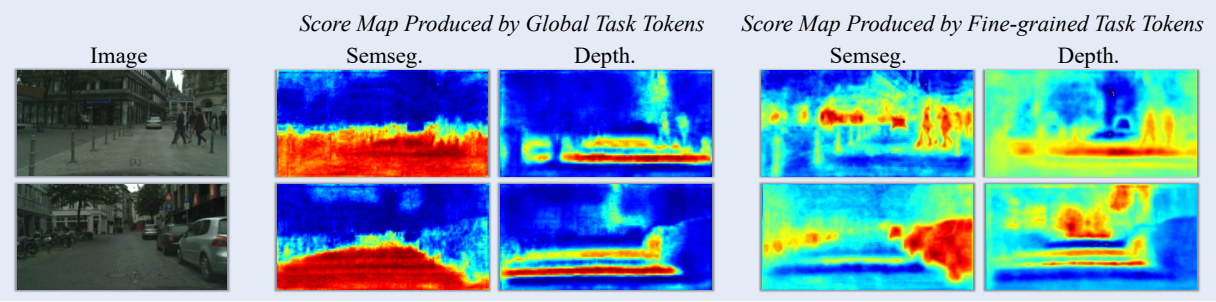


**Figure 6: Comparisons of task score maps produced by global task tokens and fine-grained task tokens on Cityscapes.**

**Table 2: Comparison on NYUD-v2 under *one-label* and *random-labels* settings.**

| Setting | Model | Semseg. *mIoU*↑ | Depth. *AbS*↓ | Normal. *mErr*↓ | $\Delta_{MTL}$ (%)↑ |
|---|---|---|---|---|---|
| One-Label | STL | 29.28 | 0.7182 | 30.1971 | - |
| | MTL *baseline* | 30.92 | 0.5982 | 31.8509 | 5.61 |
| | MTAN [33] | 30.92 | 0.6196 | 30.0278 | 6.63 |
| | XTC [28] | 33.46 | 0.5728 | 31.1492 | 10.46 |
| | JTR [41] | 31.96 | 0.5919 | 30.8000 | 8.25 |
| | Ours | **35.81** | **0.5540** | **28.5131** | **16.91** |
| Random-Labels | STL | 34.49 | 0.6272 | 27.9681 | - |
| | MTL *baseline* | 35.49 | 0.5503 | 29.9541 | 2.69 |
| | MTAN [33] | 35.96 | 0.6120 | 28.6933 | 1.36 |
| | XTC [28] | 38.11 | 0.5387 | 29.6549 | 6.19 |
| | JTR [41] | 37.08 | 0.5541 | 29.4400 | 4.63 |
| | Ours | **41.78** | **0.5177** | **27.3488** | **13.60** |

our work is the only one that achieves balanced performance gain on both tasks compared with STL.

**Table 3: Comparison on Cityscapes under *one-label* setting. "*" denotes the re-implemented results to align the settings.**

| Setting | Model | Semseg. *mIoU*↑ | Depth. *AbS*↓ | $\Delta_{MTL}$ (%)↑ |
|---|---|---|---|---|
| One-Label | STL | 69.69 | 0.0142 | - |
| | MTL *baseline* | 69.94 | 0.0159 | -5.81 |
| | MTAN [33] | 71.12 | 0.0146 | -0.38 |
| | XTC [28] | 73.23 | 0.0159 | -3.45 |
| | JTR [41] | 72.33 | 0.0163 | -5.50 |
| | DiffusionMTL (F)* [74] | 73.19 | 0.0138 | 3.92 |
| | Ours | **73.65** | **0.0135** | **5.31** |

## 4.3 Model Analysis

**Components of Hierarchical Task Tokens.** As shown in Table 4, under the one-label setting on NYUD-v2, we give an ablation study of HiTTs' key components. Generally, we analyze the role of global task tokens $\boldsymbol{\theta_i}$ and fine-grained tokens $\boldsymbol{\varphi_i}$, and core designs for token learning process, including the orthogonal embeddings

**Table 4: Investigate the effectiveness of different components on NYUD-v2 testing set under *one-label* setting. $\mathcal{L}_p^*$ represents the navie pseudo label loss in prediction space without utilizing HiTTs.**

| Method | Semseg. *mIoU*↑ | Depth. *AbS*↓ | Normal. *mErr*↓ | $\Delta_{MTL}$ (%)↑ |
|---|---|---|---|---|
| STL | 29.28 | 0.7182 | 30.1971 | - |
| MTL *baseline* | 30.92 | 0.5982 | 31.8509 | 5.61 |
| HiTTs *w/o.* OE | 27.38 | 0.6049 | 30.5904 | 2.66 |
| HiTTs *w/o.* Inter-Task Attention | 31.26 | 0.5966 | 30.2911 | 7.79 |
| HiTTs *w/o.* Intra-Task Attention | 31.44 | 0.5910 | 30.1432 | 8.42 |
| HiTTs *w/o.* $\boldsymbol{\theta_i}$ | 30.03 | 0.5823 | 30.0005 | 7.38 |
| HiTTs *w/o.* $\boldsymbol{\varphi_i}$ | 30.53 | 0.5842 | 30.0891 | 7.76 |
| HiTTs | 32.48 | 0.5844 | 30.0847 | 9.98 |
| STL *w.* $\mathcal{L}_p^*$ | 30.78 | 0.6693 | 30.2420 | 3.93 |
| MTL *w.* $\mathcal{L}_p^*$ | 33.59 | 0.5882 | 29.8174 | 11.36 |
| HiTTs *w.* $\mathcal{L}_f$ | 33.24 | 0.5708 | 29.2227 | 12.88 |
| HiTTs *w.* $\mathcal{L}_p$ | 35.22 | 0.5613 | 28.8852 | 15.49 |
| HiTTs *w.* $\mathcal{L}_p + \mathcal{L}_f$ (*full method*) | **35.81** | **0.5540** | **28.5131** | **16.91** |

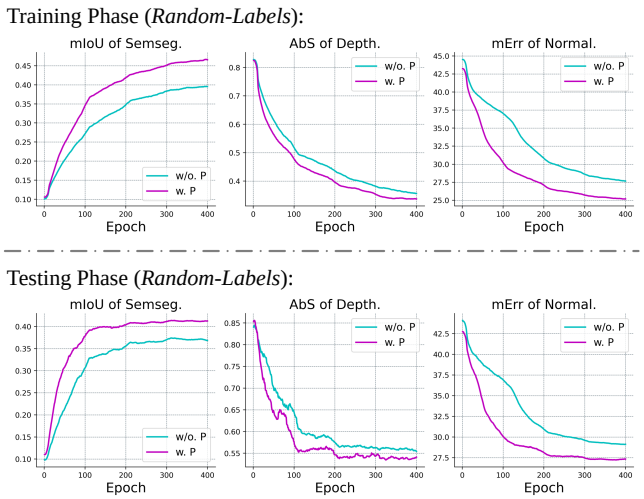


**Figure 7: Comparison of the training and testing performance on each task with and without hierarchical Pseudo Supervision (P). The model trained with pseudo supervision converges faster on both train and test splits, and gains better performance.**

(OE) (3.2), inter- and intra-task attention (3.1). The quantitative results clearly show that both hierarchies of tokens contribute to the multi-task performance and HiTTs boost the model performance by overall +4.37% $\Delta_{MTL}$ on all tasks compared with baseline. For the learning process of HiTTs, the inter- and intra-task attention both contribute to the learning process, and the orthogonal embeddings (OE) are essential for generating representative fine-grained task tokens, and without OE, the performance will significantly drop ($-7.32\%$ $\Delta_{MTL}$).

**Effect of Hierarchical Feature Supervision and Label Discovery.** To validate that our token-based label discovery is superior to the naive pseudo-labeling process, we compare $\mathcal{L}_p$ imposed on different models, including STL, MTL baselines, and ours. The HiTTs

**Table 5: Investigate the performance on labeled and unlabeled data on NYUD-v2 training set under the *one-label* setting. *GT* and *pseudo* represents the ground-truth and the pseudo supervision, respectively. Our method clearly shows effective learning on unlabeled training data.**

| Method | Supervision | | Semseg. | Depth. | Normal. |
|---|---|---|---|---|---|
| | *GT* | *Pseudo* | *mIoU*↑ | *AbS*↓ | *mErr*↓ |
| MTL *baseline* | ✓ | ✗ | 89.00 | 0.2041 | 25.9280 |
| | ✗ | ✗ | 34.31 | 0.5823 | 31.7697 |
| HiTTs | ✓ | ✗ | 86.04 | 0.3016 | 21.7911 |
| | ✗ | ✗ | 34.69 | 0.5699 | 29.8920 |
| HiTTs *w.* $\mathcal{L}_p + \mathcal{L}_f$ | ✓ | ✗ | 86.63 | 0.3173 | 20.9220 |
| | ✗ | ✓ | **37.25** | **0.5563** | **28.5169** |

perform i) effective cross-task feature-token interactions; ii) consistent label discovery in both feature and prediction space, which yields better pseudo label quality, and significantly surpasses simply applying $\mathcal{L}_p$ to STL or MTL baselines without HiTTs. We also analyze the contributions from two types of pseudo supervision losses ($\mathcal{L}_p$ and $\mathcal{L}_f$) on different hierarchies. As shown in Table 4, both methods boost multi-task performance: +7.27% $\Delta_{MTL}$ for $\mathcal{L}_f$ and +9.88% $\Delta_{MTL}$ for $\mathcal{L}_p$ compared with MTL baseline. The combination of both methods achieves better performance (+11.30% $\Delta_{MTL}$) than applying them separately, which validates the importance of consistently discovering supervision signals in both hierarchies.

**Visualization Results.** We visualize: i) The visualization of score maps produced by $\boldsymbol{\theta_i}$ and $\boldsymbol{\varphi_i}$ in Fig. 6. The visualizations reveal that score maps from global tokens provide a coarse, noisy overview and are biased towards common categories (e.g., focusing only on "road" in Cityscapes). In contrast, maps from fine-grained tokens are detailed, less noisy, and can identify smaller, less frequent objects (e.g., "vehicles" and "pedestrians"). This confirms our hierarchical structure is essential for learning representations at different levels of granularity. ii) The learning curves of metrics on every task in Fig. 7, both training and testing performance are boosted consistently on all tasks with the discovered pseudo supervision (P) on both hierarchies.

**Learning Effect on Unlabeled Data.** We also study the performance of our method on the labeled and unlabeled data separately on NYUD-v2 training set under one-label setting. As shown in Table 5, for data without labels, the model with HiTTs generalizes better on them, especially on Depth. and Normal, and adding hierarchical supervision will more significantly boost the performance on unlabeled data.

## 5 Conclusion

In this work, we propose to learn Hierarchical Task Tokens (HiTTs) for both pseudo feature supervision and label discovery under Multi-Task Partially Supervised Learning. The global task tokens are exploited for feature-token cross-task interactions and provide feature-level supervision, while the fine-grained tokens inherit knowledge from global tokens and excavate pixel pseudo labels. Extensive experimental results on partially annotated multi-task dense prediction benchmarks validate the effectiveness of our method.

## References


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

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966* (2023).

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).

[4] David Brüggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. 2021. Exploring relational context for multi-task dense prediction. In *ICCV*. 15869–15878.

[5] Yancheng Cai, Bo Zhang, Baopu Li, Tao Chen, Hongliang Yan, Jingdong Zhang, and Jiahao Xu. 2023. Rethinking cross-domain pedestrian detection: A background-focused distribution alignment framework for instance-free one-stage detectors. *IEEE transactions on image processing* 32 (2023), 4935–4950.

[6] Mang Cao, Sanping Zhou, Ye Deng, Wenli Huang, Le Wang, and Jinjun Wang. [n. d.]. MSM: Multi-Scale Mamba in Multi-Task Dense Prediction. ([n. d.]).

[7] Ruchika Chavhan, Abhinav Mehrotra, Malcolm Chadwick, Alberto Gil Ramos, Luca Morreale, Mehdi Noroozi, and Sourav Bhattacharya. 2025. Upcycling Text-to-Image Diffusion Models for Multi-Task Capabilities. *arXiv preprint arXiv:2503.11905* (2025).

[8] Linwei Chen, Ying Fu, Kaixuan Wei, Dezhi Zheng, and Felix Heide. 2023. Instance segmentation in the dark. *International Journal of Computer Vision* 131, 8 (2023), 2198–2218.

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*. 801–818.

[10] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*. PMLR, 794–803.

[11] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1290–1299.

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *CVPR*. 3213–3223. doi:10.1109/CVPR.2016.350

[13] Camille Couprie, Clément Farabet, Laurent Najman, and Yann LeCun. 2013. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572* (2013).

[14] Jean-Antoine Désidéri. 2012. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique* 350, 5-6 (2012), 313–318.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[16] David Eigen and Rob Fergus. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*. 2650–2658.

[17] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2009. The pascal visual object classes (voc) challenge. *IJCV* 88 (2009), 303–308. doi:10.1007/s11263-009-0275-4

[18] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. 2019. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *CVPR*. 3205–3214.

[19] Zhangxuan Gu, Haoxing Chen, and Zhuoer Xu. 2024. Diffusioninst: Diffusion model for instance segmentation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2730–2734.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[21] Lukas Hoyer, David Joseph Tan, Muhammad Ferjad Naeem, Luc Van Gool, and Federico Tombari. 2024. SemiVL: semi-supervised semantic segmentation with vision-language guidance. In *European Conference on Computer Vision*. Springer, 257–275.

[22] Abdullah-Al-Zubaer Imran, Chao Huang, Hui Tang, Wei Fan, Yuan Xiao, Dingjun Hao, Zhen Qian, and Demetri Terzopoulos. 2020. Partly Supervised Multitask Learning. *arXiv preprint arXiv:2005.02523* (2020).

[23] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. 2019. Label propagation for deep semi-supervised learning. In *CVPR*. 5070–5079.

[24] Maximilian Jaritz, Jiayuan Gu, and Hao Su. 2019. Multi-view pointnet for 3d scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*. 0–0.

[25] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7482–7491.

[26] Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML*, Vol. 3. 896.

[27] Bo Li, Yuchao Dai, and Mingyi He. 2018. Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. *Pattern Recognition* 83 (2018), 328–339.

[28] Wei-Hong Li, Xialei Liu, and Hakan Bilen. 2022. Learning multiple dense prediction tasks from partially annotated data. In *CVPR*. 18879–18889.

[29] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. 2019. Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*. 6936–6945.

[30] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2023. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122* (2023).

[31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems* 36 (2023), 34892–34916.

[32] Qiuhua Liu, Xuejun Liao, and Lawrence Carin. 2007. Semi-supervised multitask learning. *NIPS* 20 (2007).

[33] Shikun Liu, Edward Johns, and Andrew J Davison. 2019. End-to-end multi-task learning with attention. In *CVPR*. 1871–1880.

[34] Yuxiang Lu, Shengcao Cao, and Yu-Xiong Wang. 2024. Swiss army knife: Synergizing biases in knowledge from vision foundation models for multi-task learning. *arXiv preprint arXiv:2410.14633* (2024).

[35] Yao Lu, Soren Pirk, Jan Dlabal, Anthony Brohan, Ankita Pasad, Zhao Chen, Vincent Casser, Anelia Angelova, and Ariel Gordon. 2021. Taskology: Utilizing task relations at scale. In *CVPR*. 8700–8709.

[36] Xiangde Luo, Jieneng Chen, Tao Song, and Guotai Wang. 2021. Semi-supervised medical image segmentation through dual-task consistency. In *AAAI*, Vol. 35. 8801–8809.

[37] Huayu Mai, Rui Sun, Tianzhu Zhang, and Feng Wu. 2024. Rankmatch: Exploring the better consistency regularization for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3391–3401.

[38] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. 2019. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1851–1860.

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

[40] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *CVPR*. 3994–4003.

[41] Kento Nishi, Junsik Kim, Wanhua Li, and Hanspeter Pfister. 2024. Joint-Task Regularization for Partially Labeled Multi-Task Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16152–16162.

[42] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. 2023. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 815–824.

[43] Alessandro Pieropan, Hossein Azizpour, Atsuto Maki, et al. 2022. Dense FixMatch: a simple semi-supervised learning method for pixel-wise prediction tasks. *arXiv preprint arXiv:2210.09919* (2022).

[44] Lingyan Ran, Yali Li, Guoqiang Liang, and Yanning Zhang. 2024. Pseudo labeling methods for semi-supervised semantic segmentation: A review and future perspectives. *IEEE Transactions on Circuits and Systems for Video Technology* (2024).

[45] Tianhe Ren, Yihao Chen, Qing Jiang, Zhaoyang Zeng, Yuda Xiong, Wenlong Liu, Zhengyu Ma, Junyi Shen, Yuan Gao, Xiaoke Jiang, et al. 2024. Dino-x: A unified vision model for open-world object detection and understanding. *arXiv preprint arXiv:2411.14347* (2024).

[46] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. 2022. Mask3d: Mask transformer for 3d semantic instance segmentation. *arXiv preprint arXiv:2210.03105* (2022).

[47] Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, and Nanning Zheng. 2018. Transductive semi-supervised deep learning using min-max features. In *ECCV*. 299–315.

[48] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *ECCV*. Springer, 746–760. doi:10.1007/978-3-642-33715-4_54

[49] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *NIPS* 33 (2020), 596–608.

[50] Roger Alan Stein, Patricia A Jaques, and Joao Francisco Valiati. 2019. An analysis of hierarchical text classification using word embeddings. *Information Sciences* 471 (2019), 216–232.

[51] Aixin Sun and Ee-Peng Lim. 2001. Hierarchical text classification and evaluation. In *Proceedings 2001 IEEE International Conference on Data Mining*. IEEE, 521–528.

[52] Hui Tang and Kui Jia. 2022. Towards Discovering the Effectiveness of Moderately Confident Samples for Semi-Supervised Learning. In *CVPR*. 14658–14667.

[53] Yingjie Tang, Shou Feng, Chunhui Zhao, Yongqi Chen, Zhiyong Lv, and Weiwei Sun. 2025. A Semantic Change Detection Network Based on Boundary Detection and Task Interaction for High-Resolution Remote Sensing Images. *IEEE Transactions on Neural Networks and Learning Systems* (2025).

[54] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NIPS* 30 (2017).

[55] Yuxin Tian, Yijie Lin, Qing Ye, Jian Wang, Xi Peng, and Jiancheng Lv. 2024. UNITE: multitask learning with sufficient feature for dense prediction. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 54, 8 (2024), 5012–5024.

[56] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. 2021. Multi-task learning for dense prediction tasks: A survey. *PAMI* (2021).

[57] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. 2020. Mti-net: Multi-scale task interaction networks for multi-task learning. In *ECCV*. Springer, 527–543.

[58] Yizhou Wang, Kuan-Chuan Peng, and Yun Fu. 2025. Towards zero-shot 3d anomaly localization. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1447–1456.

[59] Yizhou Wang, Can Qin, Yue Bai, Yi Xu, Xu Ma, and Yun Fu. 2022. Making reconstruction-based method great again for video anomaly detection. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1215–1220.

[60] Yizhou Wang, Can Qin, Rongzhe Wei, Yi Xu, Yue Bai, and Yun Fu. 2022. Self-supervision meets adversarial perturbation: A novel framework for anomaly detection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4555–4559.

[61] Yizhou Wang, Can Qin, Rongzhe Wei, Yi Xu, Yue Bai, and Yun Fu. 2024. Sla ^2 p: Self-supervised anomaly detection with adversarial perturbation. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[62] Yufeng Wang, Yi-Hsuan Tsai, Wei-Chih Hung, Wenrui Ding, Shuo Liu, and Ming-Hsuan Yang. 2022. Semi-supervised multi-task learning for semantics and depth. In *WACV*. 2505–2514.

[63] Yizhou Wang, Lingzhi Zhang, Yue Bai, Mang Tik Chiu, Zhengmian Hu, Mingyuan Zhang, Qihua Dong, Yu Yin, Sohrab Amirghodsi, and Yun Fu. 2025. Cautious Next Token Prediction. *arXiv preprint arXiv:2507.03038* (2025).

[64] Yizhou Wang, Ruiyi Zhang, Haoliang Wang, Uttaran Bhattacharya, Yun Fu, and Gang Wu. 2023. Vaquita: Enhancing alignment in llm-assisted video understanding. *arXiv preprint arXiv:2312.02310* (2023).

[65] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10687–10698.

[66] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. 2018. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*. 675–684.

[67] Lihe Yang, Lei Qi, Litong Feng, Wayne Zhang, and Yinghuan Shi. 2023. Revisiting weak-to-strong consistency in semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7236–7246.

[68] Lihe Yang, Zhen Zhao, and Hengshuang Zhao. 2025. Unimatch v2: Pushing the limit of semi-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025).

[69] Siwei Yang, Hanrong Ye, and Dan Xu. 2023. Contrastive multi-task dense prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

[70] Yuqi Yang, Peng-Tao Jiang, Qibin Hou, Hao Zhang, Jinwei Chen, and Bo Li. [n. d.]. Multi-Task Dense Predictions via Unleashing the Power of Diffusion. In *The Thirteenth International Conference on Learning Representations*.

[71] Hanrong Ye and Dan Xu. 2022. Inverted Pyramid Multi-task Transformer for Dense Scene Understanding. *ECCV* (2022).

[72] Hanrong Ye and Dan Xu. 2023. TaskExpert: Dynamically Assembling Multi-Task Representations with Memorial Mixture-of-Experts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 21828–21837.

[73] Hanrong Ye and Dan Xu. 2023. TaskPrompter: Spatial-Channel Multi-Task Prompting for Dense Scene Understanding. In *ICLR*.

[74] Hanrong Ye and Dan Xu. 2024. DiffusionMTL: Learning Multi-Task Denoising Diffusion Model from Partially Annotated Data. In *CVPR*.

[75] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. 2020. Robust learning through cross-task consistency. In *CVPR*. 11197–11206.

[76] Bernhard Zeisl, Marc Pollefeys, et al. 2014. Discriminatively trained dense surface normal estimation. In *ECCV*. Springer, 468–484.

[77] Yu Zeng, Yunzhi Zhuge, Huchuan Lu, and Lihe Zhang. 2019. Joint learning of saliency detection and weakly supervised semantic segmentation. In *ICCV*. 7223–7233.

[78] Jingdong Zhang, Jiayuan Fan, Peng Ye, Bo Zhang, Hancheng Ye, Baopu Li, Yancheng Cai, and Tao Chen. 2023. Rethinking of Feature Interaction for Multi-task Learning on Dense Prediction. *arXiv preprint arXiv:2312.13514* (2023).

[79] Jingdong Zhang, Jiayuan Fan, Peng Ye, Bo Zhang, Hancheng Ye, Baopu Li, Yancheng Cai, and Tao Chen. 2025. BridgeNet: Comprehensive and Effective Feature Interactions via Bridge Feature for Multi-Task Dense Predictions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025).

[80] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. 2018. Joint task-recursive learning for semantic segmentation and depth estimation. In *ECCV*. 235–251.

[81] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. 2019. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *CVPR*. 4106–4115.

[82] Jie Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and Gongshen Liu. 2020. Hierarchy-aware global model for hierarchical text classification. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. 1106–1117.

[83] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159* (2020).

[84] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*. 289–305.

[85] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. 2019. Confidence regularized self-training. In *ICCV*. 5982–5991.

# Supplementary

In this supplementary document, we present: (i) more details about the structure of hierarchical task tokens and training pipeline, (ii) a more comprehensive explanation of experimental implementations, (iii) more quantitative and qualitative experimental results.

## A Model Details

### A.1 Training Pipeline

We will discuss utilizing the discovered supervision signals by Hierarchical Task Tokens (HiTTs) for the training process in this section. We first train the multi-task model with HiTTs on all labeled task data. Then the weights of the network are fixed, and used to produce hierarchical supervision on both feature and prediction spaces. We produce the pseudo label in an offline manner according to [65], which is labeling on clean image without data augmentation, and training on augmented images and pseudo labels to enforce consistent predictions. In [65], this method is only applied to classification tasks while we extend the utilization to general dense prediction tasks. After the pseudo labels are produced, we use them along with the ground-truth labels to jointly train the multi-task model from scratch. Additionally, we also use the pseudo feature supervision produced by this pre-trained multi-task model for feature regularization during the optimization process. Both hierarchies of the discovered supervision signals ensure that all task predictions will obtain pixel-wise supervision for multi-task optimization to gain better generalization ability on unlabeled data.

### A.2 Global Task Token Learning

In this section we are going to introduce the inter- and intra-task attention in detail. Given an RGB input $\boldsymbol{X} \in \mathbb{R}^{3\times H\times W}$, a multi-task dense prediction framework firstly produces a task-generic representation $\boldsymbol{F_s} \in \mathbb{R}^{C\times h\times w}$ through a shared encoder. Considering we have $T$ tasks, and we target decoding task features $\{\boldsymbol{F_1}, \boldsymbol{F_2}, \cdots, \boldsymbol{F_T}\}$ from $\boldsymbol{F_s}$ as well as learning representative global task tokens $\{\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \cdots, \boldsymbol{\theta_T}\}$ for each task.

As shown in Fig. 8: i) Inter-Task Learning, which aims to learn explicit global cross-task token affinities $\boldsymbol{A}$, and conduct cross-task interaction accordingly for the global token learning process. ii) Intra-task Learning, which learns task-specific information by globally conducting self-attention between task feature and token pairs.

Firstly, we flatten the shared feature $\boldsymbol{F_s}$ into feature tokens with shape $\mathbb{R}^{C\times(hw)}$, and use each global task token to query the shared feature to obtain each task feature $\boldsymbol{F_i}$ accordingly. Then, to conduct *inter-task attention*, all global task tokens are used to produce cross-task affinities that explicitly guide the learning process. The cross-task affinity map $\boldsymbol{A} \in \mathbb{R}^{T\times T}$ is calculated as:

$$\boldsymbol{A} = \text{Softmax}(\boldsymbol{Q} \times \boldsymbol{K}^\top), \tag{5}$$

where $\boldsymbol{Q}$ and $\boldsymbol{K}$ are individual linear projection of concatenated global tokens $\boldsymbol{\Theta} = [\boldsymbol{\theta_1}; \boldsymbol{\theta_2}; \cdots; \boldsymbol{\theta_T}]^\top$, in which $[\cdot]$ indicates the concatenation. After affinity matrix $\boldsymbol{A}$ is calculated, it is used to conduct affine combinations of task features and global task tokens respectively:

$$\boldsymbol{\Theta'} = \boldsymbol{A} \times \boldsymbol{\Theta}, \quad \boldsymbol{\mathcal{F}'} = \boldsymbol{A} \times \boldsymbol{\mathcal{F}}, \tag{6}$$

and similarly, $\boldsymbol{\mathcal{F}} = [\boldsymbol{F_1}; \boldsymbol{F_2}; \cdots; \boldsymbol{F_T}]^\top$, and $\boldsymbol{\Theta'}$, $\boldsymbol{\mathcal{F}'}$ represents all updated task tokens and features after the affine combinations. For each task feature $\boldsymbol{F_i}$, if it is not directly supervised by labels, the feature will be less representative and contain more noise. Thus, conducting affine combinations among all tasks ensures that the task-shared representations from labeled tasks are able to fertilize the unlabeled task features.

Afterward, the updated tokens and features with cross-task information are involved in the *intra-task attention*, where we first concatenate every corresponding task token and feature, and perform self-attention on the spatial dimension among each token-feature pair $\left[\boldsymbol{\theta_i'}; \boldsymbol{F_i'}\right] \in \mathbb{R}^{C\times(hw+1)}$. The global task tokens will further learn more specific and discriminative task representations during this process, and representative task tokens will in turn enhance the feature quality as well. Followingly, we discover pseudo feature supervision with the aid of well-learned global task tokens $\boldsymbol{\theta_i'}$.

Additionally, for multi-scale backbone features, directly fusing them ignores the various granularity of task representations maintained at different scales. Thus, for multi-scale image backbone, we further propose Multi-scale Global Task Token Learning in order to learn comprehensive multi-scale task relations. The proposed method involves *inter-task attention* separately at each scale, and then the multi-scale features and tokens are fused before *intra-task attention*. In this way, the global task tokens gain richer cross-task relations at different scales and are able to maintain stronger representations. We will illustrate this part in detail in Sec. A.4.

### A.3 Discrete Quantization and Task Losses

In Sec. 3 of the body part, we have discussed how to learn HiTTs. For continuous regression tasks, such as Depth and Normal, we first need to perform a discrete quantization of the label space to provide discriminative supervision for tokens. The goal for quantization is to assign meaningful category bins to each fine-grained token for classification. As analyzed in [4], this quantization only changes the way of predicting regression task, but does not contribute to the learning performance. For depth estimation, we follow the setting in [4, 27], and divide the range of depth values into several logarithmic bins. Our predicted task logits score $\boldsymbol{G_i'}$ is used to calculate the soft-weighted sum with each bin and produce final task predictions accordingly. For surface normal estimation, we follow [4, 76] and use K-means to learn several unit normal vectors, which serve as clustering centers, and they are also used to generate predictions with $\boldsymbol{G_i'}$. This process can be expressed as:

$$\hat{\boldsymbol{Y}}_{\boldsymbol{i}} = \text{Sum}\left(\boldsymbol{c_i}^\top \times \text{Softmax}(\boldsymbol{G_i'})\right), \tag{7}$$

where $\boldsymbol{c_i} \in \mathbb{R}^{C_P}$ represents the center of each bin. In our experiments, the numbers of depth bins and normal cluster centers on NYUD-v2 are 30 and 20, respectively. For Cityscapes, we consider 100 depth bins as the Cityscapes dataset is captured from outdoor scenarios and have more significant changes in depth. For PASCAL-Context, we select 40 different unit verctors uniformly from the space to serve as normal cluster centers.

To supervise the task predictions, we can directly impose regression losses on $\hat{\boldsymbol{Y}}_{\boldsymbol{i}}$ for Depth. and Normal:

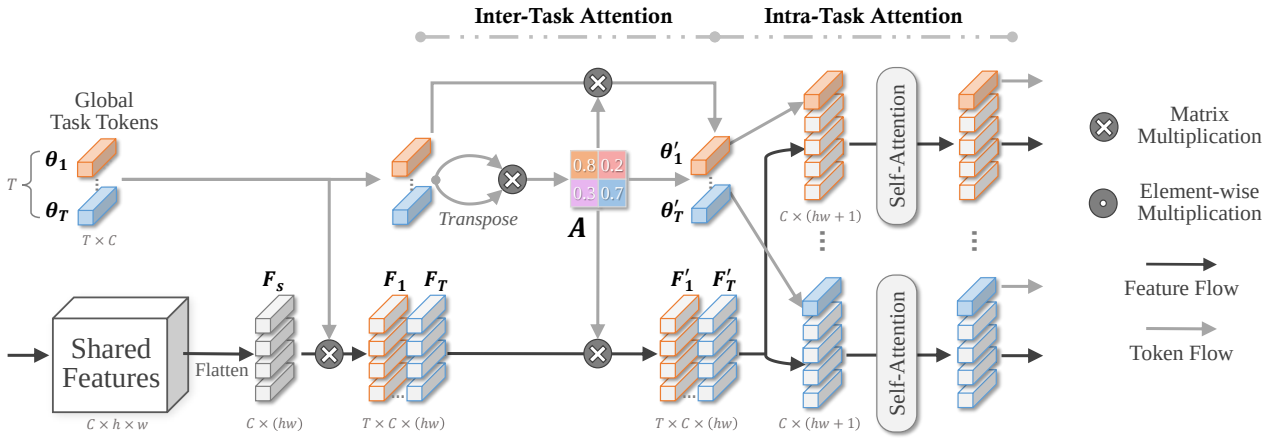


**Figure 8: Illustration of detail designs of: (i) Inter-Task Attention: predicting cross-task token affinities $A$ from the global task tokens. (ii) Intra-Task Attention: conducting self-attention between task features and tokens.**

$$\mathrm{L}_{\mathrm{i}}^{\mathrm{reg}}\left(\hat{Y}_i, Y_i\right), \tag{8}$$

where $\mathrm{L}_{\mathrm{i}}^{\mathrm{reg}}(\cdot)$ can be $\mathrm{L}_1$ loss or Angle loss for Depth. or Normal. respectively. Furthermore, in order to gain more discriminative task category information for each token, we also impose classification loss on $G_i'$. We first extract the corresponding one-hot label from $Y_i$ by $c_i$, denoting as $Y_i^{oh}$, and the loss can be written as:

$$\mathrm{L}_{\mathrm{i}}^{\mathrm{cls}}\left(G_i, Y_i^{oh}\right) = -Y_i^{oh} \cdot \log\left(\mathrm{Softmax}(G_i')\right), \tag{9}$$

and the overall **mixture loss** for each task can be written as:

$$\mathrm{L}_{\mathrm{i}}(\cdot) = \lambda_i \mathrm{L}_{\mathrm{i}}^{\mathrm{reg}}\left(\hat{Y}_i, Y_i\right) + \mathrm{L}_{\mathrm{i}}^{\mathrm{cls}}\left(G_i', Y_i^{oh}\right), \tag{10}$$

where $\lambda_i = 0.1$ if task $i$ is a regression task itself, otherwise $\lambda_i = 0$.

## A.4 Multi-scale Global Task Token Learning

For the implementations on NYUD-v2 and Cityscapes, we use the SegNet [1] to serve as the shared backbone, which produces single-scale shared features for per-task decoding. However, on PASCAL-Context, we use ResNet-18 [20] as the shared backbone, which can produce multi-scale shared features for decoding. Directly fusing them ignores the various task information maintained in different scales. Thus, we propose to learn the global task tokens on different scales in order to learn more comprehensive task relations.

As shown in Fig. 9, compared with the single-scale global task token learning process, the multi-scale process involves *inter-task learning* separately on each scale, and then the multi-scale features and tokens are fused before *intra-task learning*. The multi-scale backbone features $\left\{F_s^{(j)}\right\}, j = 0, 1, 2, 3$ are first flattened on each scale, and each global task token $\theta_i$ is projected by a linear layer to produce multi-scale tokens:

$$\theta_i^{(j)} = W_i^{s\to m(j)} \times \theta_i, \quad j = 0, 1, 2, 3, \tag{11}$$

where $W_i^{s\to m(j)}$ Then, for every feature $F_s^{(j)}$ and token $\theta_i^{(j)}$ on scale $j$ and task $i$, we query $F_s^{(j)}$ to obtain task features $\left\{F_i^{(j)}\right\}, i = 1, 2, \cdots, T; j = 0, 1, 2, 3$. After that, on each scale, we concatenate features and tokens from every task for *Inter-task Learning* similar to Sec. 3.1:

$$\mathcal{F}^{(j)} = \left[F_1^{(j)}; F_2^{(j)}; \cdots; F_T^{(j)}\right]^\top, \tag{12}$$

$$\Theta^{(j)} = \left[\theta_1^{(j)}; \theta_2^{(j)}; \cdots; \theta_T^{(j)}\right]^\top. \tag{13}$$

Subsequently, $\mathcal{F}^{(j)}$ and $\Theta^{(j)}$ are used for inter-task learning on each scale. We denote the features and tokens after intra-task learning as $\mathcal{F}'^{(j)}$ and $\Theta'^{(j)}$. We fused them to share cross-task information on each scale:

$$F_i' = \mathrm{Conv}_{\mathrm{i}}^{1\times1}\left(\left[F_i'^{(0)}; F_i'^{(1)}; F_i'^{(2)}; F_i'^{(3)}\right]\right), \tag{14}$$

$$\theta_i' = \sum_{j=0}^{3}\left(W_i^{m\to s(j)} \times \theta_i'^{(j)}\right). \tag{15}$$

Finally, on each task $i$, the updated task features $F_i'$ and global task tokens $\theta_i'$ are used for *Intra-task Learning*. The process is the same as Sec. 3.1.

In this way, we achieve learning global task tokens on multi-scale task features, which gains richer cross-task relations on different scales and maintains stronger representations in the global task tokens.

## A.5 Broader Applicability and Future Directions

The hierarchical design of our HiTTs is highly generalizable and not limited to the currently studied tasks. Its principles can be readily extended to other fundamental computer vision tasks, such as object detection [3, 5, 45, 59, 83] and instance segmentation [8, 11, 19, 46], which fundamentally rely on rich, multi-scale pixel representations. The experiment on the PASCAL-Context dataset partially demonstrates this potential, where HiTTs effectively learns from both coarse-grained semantic labels and fine-grained human part annotations simultaneously, showcasing its robust joint-learning capabilities across different granularities.

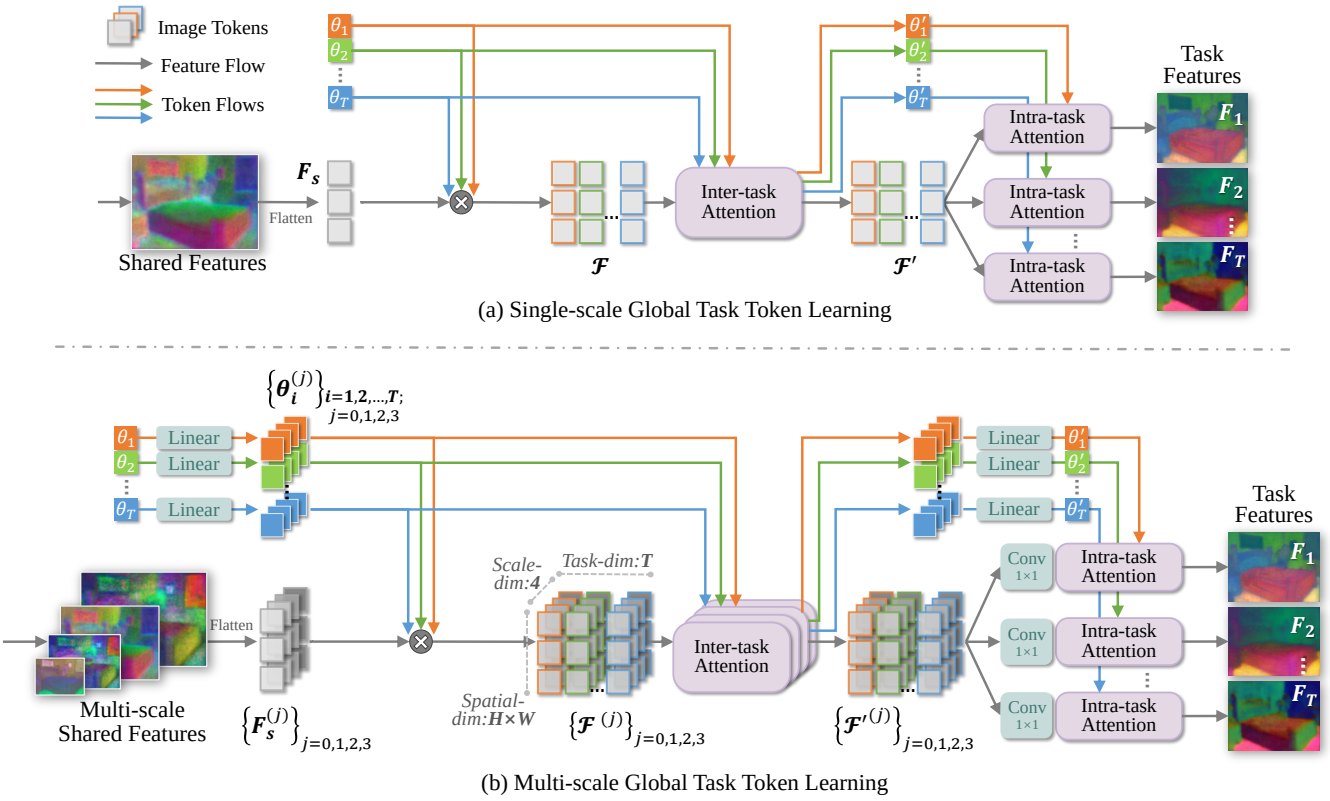


**Figure 9: Illustrations of *Single-scale* Global Task Token Learning (a) and *Multi-scale* Global Task Token Learning (b). For the multi-scale features produced by the shared backbone, we use linear layers to produce corresponding task tokens for each scale. Then, in each scale, we query task features and conduct *inter-task attention* to gain multi-task multi-scale representations. The multi-scale features and tokens are fused before *intra-task attention* to transfer cross-scale information.**

Furthermore, the core concepts of HiTTs extend beyond the visual domain. Its ability to build cross-task relations upon multi-level representations makes it a promising approach for tasks like hierarchical text classification [50, 51, 60, 61, 82] or 3D scene-level parsing [24, 42, 58]. Moreover, the unified token-feature interaction mechanism, based on the Transformer architecture, is inherently compatible with diverse data types, paving the way for future exploration in multi-modal learning environments [2, 30, 31, 63, 64]. This adaptability underscores the broad potential of our hierarchical token-based approach for complex multi-task and multi-modal problems.

# B Implementation Details

## B.1 Dataset

**PASCAL-Context.** PASCAL-Context [17] contains 4998 and 5105 images for training and testing respectively, which also have pixel-level annotations for semantic segmentation, human-parts segmentation and semantic edge detection. Additionally, we also consider surface normal estimation and saliency detection distilled by [38]. We use Adam optimizer with learning rate $2 \times 10^{-5}$, and weight decay $1 \times 10^{-6}$, and train for 100 epochs with batch size 6. We update the learning rate with polynomial strategy and $\gamma = 0.9$ for the power factor.

**NYUD-v2.** NYUD-v2 [48] contains 795 and 654 RGB-D indoor scene images for training and testing respectively. We use the 13-class semantic annotations which is defined in [13], the truth depth annotations recorded by Microsoft Kinect depth camera, and surface normal annotations which are produced in [16]. Following the setting in [28, 33], we use $288 \times 384$ image resolution to speed up training. We use Adam optimizer with a learning rate of $1 \times 10^{-4}$, and train for 400 epochs with batch size 8. We update the learning rate every 100 epoch with $\gamma = 0.5$ as the multiplying factor.

**Cityscapes.** Cityscapes [12] contains 2975 and 500 street-view images for training and testing respectively. We used the projected 7-class semantic annotations from [33], and disparity maps to serve as depth annotations. Following the setting in [28, 33], we use $128 \times 256$ image resolution to speed up training. The optimizer and learning rate scheduler are set as the same as NYUD-v2.

## B.2 Data Preprocessing

We follow the setting of [28, 33, 56, 74] to process training data. For NYUD-v2 and Cityscapes, we use random scaling, cropping and horizon flipping for data augmentation following [28, 33]. For PASCAL-Context, we follow [74] and use random scaling, cropping, horizon flipping and photometric distortion for data augmentation.

Apart from the two partial settings (**one-label** and **random-labels**) we mentioned in Sec.4.1, we also consider two extra settings: **(iii) full-labels**: each image has labels on every task; **(iv) few-shot:** one task has only very few labels while other tasks are fully supervised. Both of the extra settings will furthermore show the effectiveness of our method.

## B.3 Model Setting

We use SegNet [1] as the image backbone for our experiments on NYUD-v2 and Cityscapes; and we use ResNet-18 as the image backbone, Atrous Spatial Pyramid Pooling (ASPP) [9] as the task-specific decoding heads. For the threshold $\tau_i$ which selects binary mask $\boldsymbol{M}_i^P$, we use {Semseg: 0.9, Depth: 0.45, Normal: 0.6 } for NYUD-v2 one-label setting, {Semseg: 0.9, Depth: 0.7, Normal: 0.7} for NYUD-v2 random-labels setting. {Semseg: 0.9, Depth: 0.5} for Cityscapes one-label setting, and {Semseg: 0.9, Parsing: 0.85, Normal: 0.7, Sal: 0.7, Edge: 0.9} for PASCAL-Context one-label and random-labels settings.

## B.4 Evaluation Metrics

We have briefly introduced our evaluation metrics for multiple dense prediction tasks in Sec.4.1. We provide a more detailed description as follows: (i) mIoU: mean intersection over union; (ii) pAcc: per-pixel accuracy; (iii) AbS or AbR: absolute error or absolute-relative error; (iv) rmse: root mean square error (for Normal. we calculate the mean square error of the predicted angles with the ground-truths); (v) mErr: mean of angle error; (vi) odsF: optimal dataset F-measure [39]; (vii) threshold: for surface normal estimation, we calculate the proportion of pixels with angle error smaller than three thresholds $\eta \in \{11.25^\circ, 22.50^\circ, 30^\circ\}$.

Additionally, to better evaluate the proposed method, we also consider using $\Delta_{MTL}$ proposed by [56] to evaluate the overall improvement of the multi-task performances of all the tasks, which is defined as:

$$\Delta_{MTL} = \sum_{N} (-1)^{l_t} \left(M_{m,t} - M_{s,t}\right) / M_{s,t} \tag{16}$$

where $l_t = 1$ if a lower evaluation value indicates a better performance measurement of $M_t$ for task $t$, and $l_t = 0$ if a higher value is better. Footnote $s$ and $m$ represent the performance of the single-task learning and the multi-task learning respectively. We will show experimental results with all of these metrics to further show the effectiveness of our method.

## B.5 More Quantitative Results

### B.5.1 *State-of-the-art Comparison.*

**Comparison on NYUD-v2.** We compare our method with [28, 33] on NYUD-v2 under both the one-label and random-labels settings, and the quantitative results are shown in Table 6. XTC [28] is the first work designed for partially annotated multi-task dense prediction. MTAN [28] is an attention-based MTL network designed for the fully supervised setting, and we train it with our setup. The quantitative results show that our method surpasses them by a large margin on all the metrics of the three tasks. More specifically, ours achieves +9.63% $\Delta_{MTL}$ and +8.62% $\Delta_{MTL}$ compared with [28] under the two partial-label settings, respectively.

Our HiTTs can also be applied on full-labels setting, which utilizes cross-task relations and task-token interactions to fertilize the multi-task learning process. We compare with some of the recent works, including multi-task interaction works like MTAN [33], X-Task [75] and CCR [69]; and multi-task loss weighting strategies like Uncertainty [25], GradNorm [10], MGDA [14] and DWA [33]. As shown in Table 7, our method still clearly surpasses all of the SOTA works (13.29% $\Delta_{MTL}$ overall), indicating the effective cross-task interaction brought by HiTTs. Additionally, with the aid of feature-level supervision loss $\mathcal{L}_f$, which is supported by global task tokens, our method can achieve 14.64% $\Delta_{MTL}$ overall on the three tasks.

**Comparison on Cityscapes.** We also compare our results with [28, 33] on Cityscapes, under the *one-label* setting with both Semseg. and Depth. tasks. As shown in Table 8, our method achieves SOTA performance on both tasks, and significantly better performance on Depth, resulting in an average gain of +6.98% in terms of $\Delta_{MTL}$. Additionally, it's worth mentioning that our work is the only one that achieves balanced performance gain on both tasks compared with STL.

### B.5.2 *Model Analysis.*

**Effect of Hierarchical Task Tokens.** As shown in Table 9, under the one-label setting on NYUD-v2, we give a more-detailed ablation study on the key components of hierarchical task tokens (HiTTs), including both hierarchies of the task tokens, and the inter-task (Inter) and intra-task (Intra) learning of global task tokens, and orthogonal embeddings (OE) of fine-grained task tokens, and two types of pseudo supervision losses ($\mathcal{L}_p$ for pseudo label loss and $\mathcal{L}_f$ for feature supervision loss) on different hierarchy. The quantitative results clearly show that every component contributes to the multi-task performance on all the metrics and on all the tasks over the MTL baseline. HiTTs boost the model performance by conducting cross-task interaction and encouraging high-confidence predictions, and leads to an overall +9.64% $\Delta_{MTL}$ on all tasks compared with baseline. However, for the learning process of HiTTs, the orthogonal embeddings (OE) are essential for generating representative fine-grained task tokens, and without OE, the performance will significantly drop (−4.38% $\Delta_{MTL}$), especially on Semseg. which requires more discriminative category information. The inter-task and intra-task learning processes are also important since without either of them, the learning of cross-task relations and task representations will be affected, resulting in −2.00% $\Delta_{MTL}$ and −1.28% $\Delta_{MTL}$ performance after removing them respectively from the learning process of HiTTs.

**Effect of Hierarchical Feature Supervision and Label Discovery.** We analyze the contributions from both feature-level and prediction-level supervision with all of the metrics, as shown in Table 9, both methods boost multi-task performance, and $\mathcal{L}_p$ contributes more since fine-grained task tokens contain more specific and discriminative task information and are directly involved in the formation process of task predictions. The combination of both methods achieves better performance than applying them separately, which validates the importance of consistently discovering supervision signals in both hierarchies. Comparing $\mathcal{L}_p$ imposed on different models, including STL, MTL baselines, and our HiTTs, our token-based pseudo-label discovery is much better. Since MTL

**Table 6: Comparison on NYUD-v2 under *one-label* and *random-labels* settings. Our method shows clear performance gain over three tasks, which is consistent with the visualization results.**

| Setting | Model | Semseg. | | Depth. | | Normal. | | | | | $\Delta_{MTL}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *mIoU*↑ | *pAcc*↑ | *AbS*↓ | *rmse*↓ | *mErr*↓ | *rmse*↓ | $\eta_1$↑ | $\eta_2$↑ | $\eta_3$↑ | (%)↑ |
| One-Label | STL | 29.28 | 55.41 | 0.7182 | 1.0151 | 30.1971 | 37.7115 | 23.1532 | 46.4046 | 58.5216 | - |
| | MTL *baseline* | 30.92 | 58.23 | 0.5982 | 0.8544 | 31.8509 | 38.6313 | 19.7083 | 41.2614 | 53.6381 | 0.11 |
| | MTAN [33] | 30.92 | 57.14 | 0.6196 | 0.8477 | 30.0278 | 36.7808 | 21.4199 | 44.7805 | 57.5720 | 3.26 |
| | XTC [28] | 33.46 | 60.95 | 0.5728 | 0.8056 | 31.1492 | 37.8211 | 19.8410 | 42.2268 | 54.9997 | 3.60 |
| | Ours | **35.81** | **63.22** | **0.5540** | **0.7939** | **28.5131** | **36.1738** | **26.4985** | **50.2357** | **61.8343** | **13.23** |
| Random-Labels | STL | 34.49 | 60.52 | 0.6272 | 0.8824 | 27.9681 | 34.9293 | 24.6011 | 49.7888 | 62.4425 | - |
| | MTL *baseline* | 35.49 | 61.81 | 0.5503 | 0.7874 | 29.9541 | 36.7726 | 21.6933 | 45.0412 | 57.7516 | -1.47 |
| | MTAN [33] | 35.96 | 61.64 | 0.6120 | 0.8272 | 28.6933 | 35.3528 | 23.0253 | 47.2287 | 60.1113 | -0.48 |
| | XTC [28] | 38.11 | 64.37 | 0.5387 | 0.7755 | 29.6549 | 36.3992 | 21.7058 | 45.4801 | 58.4236 | 0.66 |
| | Ours | **41.78** | **66.50** | **0.5177** | **0.7472** | **27.3488** | **34.6820** | **27.1619** | **51.8924** | **63.7670** | **9.28** |

**Table 7: Comparison on NYUD-v2 under *full-labels* settings. Our method achieves significantly better performance compared with SoTA multi-task learning works on all of the three tasks.**

| Model | Semseg. *mIoU*↑ | Depth. *AbS*↓ | Normal. *mErr*↓ | $\Delta_{MTL}$ (%)↑ |
|---|---|---|---|---|
| STL | 37.45 | 0.6079 | 25.94 | - |
| MTL *baseline* | 36.95 | 0.5510 | 29.51 | -1.91 |
| MTAN [33] | 39.39 | 0.5696 | 28.89 | 0.03 |
| X-Task [75] | 38.91 | 0.5342 | 29.94 | 0.20 |
| Uncertainty [25] | 36.46 | 0.5376 | 27.58 | 0.87 |
| GradNorm [10] | 37.19 | 0.5775 | 28.51 | -1.87 |
| MGDA [14] | 38.65 | 0.5572 | 28.89 | 0.06 |
| DWA [33] | 36.46 | 0.5429 | 29.45 | -1.83 |
| XTC [28] | 41.00 | 0.5148 | 28.58 | 4.87 |
| XTC+Uncertainty [25] | 41.09 | 0.5090 | 26.78 | 7.58 |
| CCR [69] | 43.09 | 0.4894 | 27.87 | 9.04 |
| Ours (HiTTs) | 44.32 | 0.4813 | 25.76 | 13.29 |
| Ours (HiTTs+$\mathcal{L}_f$) | **45.47** | **0.4763** | **25.72** | **14.64** |

**Table 8: Comparison on Cityscapes under *one-label* setting.**

| Setting | Model | Semseg. | | Depth. | | $\Delta_{MTL}$ |
|---|---|---|---|---|---|---|
| | | *mIoU*↑ | *pAcc*↑ | *AbS*↓ | *rmse*↓ | (%)↑ |
| One-Label | STL | 69.69 | 91.91 | 0.0142 | 0.0271 | - |
| | MTL *baseline* | 69.94 | 91.62 | 0.0159 | 0.0292 | -4.92 |
| | MTAN [33] | 71.12 | 92.35 | 0.0146 | 0.0278 | -0.72 |
| | XTC [28] | 73.23 | 92.73 | 0.0159 | 0.0293 | -3.53 |
| | Ours | **73.65** | **92.81** | **0.0135** | **0.0265** | **3.45** |

shares a backbone that learns stronger representations on all tasks, MTL produces pseudo labels with better quality and surpasses STL a lot in performance (+3.86% $\Delta_{MTL}$). Our HiTTs perform i) consistent label discovery in both feature and prediction space; ii) effective cross-task feature-token interactions, which furthermore enhance the quality of pseudo labels, and bring extra +4.68% $\Delta_{MTL}$ overall.

We also study the performance of our method on the labeled and unlabeled data separately on NYUD-v2 training set under the one-label setting. As shown in Table 10, for data without labels, the model with HiTTs generalizes better on them, especially on Depth. and Normal, and adding hierarchical supervision will more significantly boost the performance of unlabeled data.

**Effect of Cross-Task Interactions.** To further show the effect of cross-task learning brought by HiTTs, we develop new **few-shot** settings, under which one task has only a few labels while other tasks are fully labeled. We apply this setting respectively on the three tasks of NYUD-v2, namely **few-shot-semseg**, **few-shot-depth** and **few-shot-normal**. For each few-shot task, we have 10 shots for the model to learn. As shown in Table 11, we only show the performance of the few-shot tasks in the table, and due to the lack of label supervision, the STL performs poorly on each few-shot task: 5.80 *mIoU* on Semseg, 0.9633 *AbS* on Depth, and 47.5281 *mErr* on Normal. Benefiting from the sharing backbone, MTL *baseline* performs much better, since the backbone can be fully supervised on the other two tasks, and gain stronger representations from other tasks. With the aid of HiTTs, the multi-task model can achieve an extra performance gain, since the cross-task interactions brought by intra-task learning can fertilize the unlabeled tasks in the decoding stage, which introduces more task-relevant information and discriminative representations to task features without label supervision. The gain brought by HiTTs is +7.76% on Semseg, +14.86% on Depth, and +0.49% on Normal respectively. Additionally, if we add the pseudo supervision signals to aid the learning process, the performance will be further improved: +19.82% on Semseg, +19.63% on Depth, and +3.85% on Normal compared with MTL *baseline*.

**Analysis of Multi-Scale Global Task Token Learning.** As we illustrated in Sec. A.4, we adopt multi-scale global task learning with ResNet-18 backbone on PASCAL-Context. To validate the effectiveness of learning global task tokens on multi-scale features, we compare the performance of Global Task Token Learning with single-scale and multi-scale backbone features respectively in Table 12. As shown in the table, HiTTs with single-scale (SS) Global Token Learning surpass the MTL *baseline* on both one-label and random-labels settings, with overall +0.60% $\Delta_{MTL}$ and +0.97% $\Delta_{MTL}$ on all tasks respectively, and the multi-scale (MS) Global Token Learning further enhances the performance to +2.84% $\Delta_{MTL}$ and +3.33% $\Delta_{MTL}$ on all tasks, which indicates the effectiveness of applying global token learning on multi-scale features.

**Analysis of Threshold Hyperparameter $\tau_i$.** We discuss the effect of the confidence threshold hyperparameter, $\tau_i$. This threshold is chosen to ensure that high-confidence pixel predictions are masked

**Table 9: Investigate the effectiveness of different components on NYUD-v2 testing set under *one-label* setting.**

| Method | Semseg. | | Depth. | | Normal. | | | | | $\Delta_{MTL}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | *mIoU*↑ | *pAcc*↑ | *AbS*↓ | *rmse*↓ | *mErr*↓ | *rmse*↓ | $\eta_1$↑ | $\eta_2$↑ | $\eta_3$↑ | (%)↑ |
| STL | 29.28 | 55.41 | 0.7182 | 1.0151 | 30.1971 | 37.7115 | 23.1532 | 46.4046 | 58.5216 | - |
| MTL *baseline* | 30.92 | 58.23 | 0.5982 | 0.8544 | 31.8509 | 38.6313 | 19.7083 | 41.2614 | 53.6381 | 0.11 |
| +HiTTs *w/o.* OE | 27.38 | 55.08 | 0.6049 | 0.8626 | 30.5904 | 38.1046 | 23.5233 | 45.7012 | 57.1902 | 2.13 |
| +HiTTs *w/o.* Inter | 31.26 | 57.99 | 0.5966 | 0.8592 | 30.2911 | 37.7714 | 22.7802 | 46.2590 | 58.0880 | 4.51 |
| +HiTTs *w/o.* Intra | 31.44 | 58.50 | 0.5910 | 0.8533 | 30.1432 | 37.6719 | 23.3251 | 46.4354 | 58.1508 | 5.23 |
| +HiTTs *w/o.* $\boldsymbol{\theta_i}$ | 30.03 | 58.21 | 0.5823 | 0.8389 | 30.0005 | 37.3750 | 23.4160 | 46.1824 | 58.2909 | 5.08 |
| +HiTTs *w/o.* $\boldsymbol{\varphi_i}$ | 30.53 | 57.18 | 0.5842 | 0.8565 | 30.0891 | 37.4465 | 23.2297 | 45.9603 | 58.0509 | 4.60 |
| +HiTTs | 32.48 | 59.61 | 0.5844 | 0.8382 | 30.0847 | 37.5827 | 23.9975 | 46.4790 | 58.2146 | 6.51 |
| STL *w.* $\mathcal{L}_p$ | 30.78 | 58.94 | 0.6693 | 0.9362 | 30.2420 | 37.8601 | 23.5830 | 46.4743 | 58.3739 | 3.03 |
| MTL *w.* $\mathcal{L}_p$ | 33.59 | 61.79 | 0.5882 | 0.8554 | 29.8174 | 36.9781 | 23.2875 | 45.8803 | 58.1061 | 6.89 |
| +HiTTs *w.* $\mathcal{L}_f$ | 33.24 | 60.74 | 0.5708 | 0.8200 | 29.2227 | 36.9305 | 25.7968 | 48.8173 | 60.2608 | 9.75 |
| +HiTTs *w.* $\mathcal{L}_p$ | 35.22 | 62.93 | 0.5613 | 0.8014 | 28.8852 | 36.4316 | 25.3873 | 49.1251 | 60.9806 | 11.57 |
| +HiTTs *w.* $\mathcal{L}_p + \mathcal{L}_f$ | **35.81** | **63.22** | **0.5540** | **0.7939** | **28.5131** | **36.1738** | **26.4985** | **50.2357** | **61.8343** | **13.23** |

**Table 10: Investigate the performance on labeled and unlabeled data of NYUD-v2 training set under *one-label* setting.**

| Method | Supervision | | Semseg. | | Depth. | | Normal. | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | *GT* | *Pseudo* | *mIoU*↑ | *pAcc*↑ | *AbS*↓ | *rmse*↓ | *mErr*↓ | *rmse*↓ | $\eta_1$↑ | $\eta_2$↑ | $\eta_3$↑ |
| MTL *baseline* | ✓ | ✗ | 89.00 | 96.04 | 0.2041 | 0.3434 | 25.9280 | 32.0824 | 25.8276 | 52.0758 | 65.4135 |
| | ✗ | ✗ | 34.31 | 61.56 | 0.5823 | 0.8375 | 31.7697 | 38.5071 | 19.5042 | 41.2401 | 53.8490 |
| HiTTs | ✓ | ✗ | 86.04 | 95.00 | 0.3016 | 0.4625 | 21.7911 | 28.9368 | 38.8125 | 63.6280 | 73.8326 |
| | ✗ | ✗ | 34.69 | 61.48 | 0.5699 | 0.8319 | 29.8920 | 37.3331 | 23.9788 | 46.4809 | 58.4659 |
| HiTTs *w.* $\mathcal{L}_p + \mathcal{L}_f$ | ✓ | ✗ | 86.63 | 94.89 | 0.3173 | 0.4770 | 20.9220 | 28.1260 | 41.2846 | 66.0712 | 75.6904 |
| | ✗ | ✓ | **37.25** | **63.72** | **0.5563** | **0.8074** | **28.5169** | **36.1536** | **26.4528** | **50.0344** | **61.5778** |

**Table 11: Investigate the cross-task learning effect on NYUD-v2 under the *few-shot* setting.**

| Method | Few-Shot-Semseg | | Few-Shot-Depth | | Few-Shot-Normal | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | *mIoU*↑ | *pAcc*↑ | *AbS*↓ | *rmse*↓ | *mErr*↓ | *rmse*↓ | $\eta_1$↑ | $\eta_2$↑ | $\eta_3$↑ |
| STL | 5.80 | 26.06 | 0.9533 | 1.2907 | 47.5281 | 53.6422 | 5.4343 | 17.8888 | 27.9915 |
| MTL *baseline* | 16.75 | 41.01 | 0.9165 | 1.2968 | 40.0456 | **46.3370** | 12.0348 | 26.8520 | 37.2863 |
| HiTTs | 18.05 | 44.69 | 0.7803 | 1.1272 | 39.8508 | 47.1113 | 14.1108 | 29.4719 | 39.5924 |
| HiTTs *w.* $\mathcal{L}_p + \mathcal{L}_f$ | **20.07** | **45.62** | **0.7366** | **1.0206** | **38.5029** | 46.9907 | **17.1082** | **34.7171** | **45.1765** |

**Table 12: Comparison of HiTTs with *Single-scale* (SS) and *Multi-scale* (MS) Global Task Token Learning on PASCAL-Context under the *one-label* and *random-labels* setting.**

| Setting | Model | Semseg. *mIoU*↑ | Parsing. *mIoU*↑ | Norm. *mErr*↓ | Sal. *mIoU*↑ | Edge. *odsF* ↑ | $\Delta_{MTL}$ (%)↑ |
|---|---|---|---|---|---|---|---|
| One-Label | STL | 47.7 | 56.2 | 16.0 | 61.9 | 64.0 | - |
| | MTL *baseline* | 48.4 | 55.1 | 16.0 | 61.6 | 66.5 | 0.59 |
| | HiTTs (SS) | 51.0 | 54.7 | 16.2 | 61.7 | 66.1 | 1.19 |
| | HiTTs (MS) | 52.3 | 56.2 | 15.8 | 62.0 | 67.9 | 3.43 |
| Random-Labels | STL | 60.9 | 55.3 | 14.7 | 64.8 | 66.8 | - |
| | MTL *baseline* | 58.4 | 55.3 | 16.0 | 63.9 | 67.8 | -2.57 |
| | HiTTs (SS) | 59.1 | 53.4 | 15.0 | 64.1 | 67.8 | -1.60 |
| | HiTTs (MS) | 60.3 | 55.3 | 14.7 | 64.6 | 70.2 | 0.76 |

out to serve as pseudo-labels. To analyze the sensitivity of our method to this hyperparameter, we conduct an ablation study on the NYUD-v2 dataset under the *one-label* setting. As shown in Table 13, we vary $\tau_i$ across a wide range from 0.3 to 0.95. The results demonstrate that the performance on all tasks remains remarkably stable with only minor fluctuations. Furthermore, the overall multi-task learning gain ($\Delta_{MTL}$) is consistently and significantly positive across all tested values. This demonstrates that our method is not sensitive to the choice of $\tau_i$, showcasing its robustness.

**Comparison under DiffusionMTL [74] settings on NYUD-v2.** Since DiffusionMTL [74] adopts different experimental settings from our main experiments, we re-implement our method under the specific setup of DiffusionMTL, using a ResNet-18 backbone to ensure a fair comparison. The quantitative results on NYUD-v2 under both the *one-label* and *random-labels* settings are shown in Table 15. As the results indicate, our method significantly outperforms DiffusionMTL across all metrics in the *one-label* setting. More specifically, compared to the best-performing DiffusionMTL variant, our method achieves a +6.26% higher $\Delta_m$ in the *one-label* setting and a +2.09% higher $\Delta_m$ in the *random-labels* setting. This demonstrates the effectiveness and superior performance of our approach when compared directly with other methods.

**Implementations with Different Backbones.** To demonstrate that our method can be flexibly implemented on different image backbones, we perform additional experiments implementing our

**Table 15: Comparison on NYUD-v2 under the settings of DiffusionMTL [74]. (P) and (F) represent the Prediction Diffusion and Feature Diffusion modes for DiffusionMTL [74]. "*" denotes the re-implemented results to align the settings. Our method performance outperforms previous methods.**

| Setting | Model | Semseg. *mIoU* ↑ | Depth. *absErr* ↓ | Normal. *mErr* ↓ | $\Delta_m$ (%) ↑ |
|---|---|---|---|---|---|
| One-Label | STL | 45.28 | 0.4802 | 25.93 | - |
| | MTL *baseline* | 43.92 | 0.5138 | 26.44 | -3.99 |
| | SS [28] | 27.52 | 0.6499 | 33.58 | - |
| | XTC [28] | 30.36 | 0.6088 | 32.08 | - |
| | XTC* [28] | 43.97 | 0.5140 | 26.30 | -3.79 |
| | DiffusionMTL (P) [74] | 44.97 | 0.5137 | 26.17 | -2.86 |
| | DiffusionMTL (F) [74] | 44.47 | 0.5059 | 25.84 | -2.27 |
| | Ours* | **47.30** | **0.4539** | **25.40** | **3.99** |
| Random-Labels | STL | 48.25 | 0.4792 | 24.65 | - |
| | MTL *baseline* | 45.93 | 0.4839 | 25.53 | -3.12 |
| | SS [28] | 29.50 | 0.6224 | 33.31 | - |
| | XTC [28] | 34.26 | 0.5787 | 31.06 | - |
| | XTC* [28] | 46.03 | 0.4811 | 25.97 | -3.44 |
| | DiffusionMTL (P) [74] | 47.44 | 0.4803 | 25.26 | -1.45 |
| | DiffusionMTL (F) [74] | 46.82 | 0.4743 | **24.75** | -0.77 |
| | Ours* | **47.73** | **0.4510** | 24.86 | **1.32** |

**Table 16: Ablation study of our method with different backbones on NYUD-v2. The results show that our method is flexible and achieves significant gains when paired with a more powerful ViT backbone.**

| Setting | Backbone | Semseg. *mIoU* ↑ | Depth. *absErr* ↓ | Normal. *mErr* ↓ |
|---|---|---|---|---|
| One-Label | ResNet-18 | 47.30 | 0.4539 | 25.40 |
| | ViT-base | **58.38** | **0.3740** | **23.65** |
| Random-Labels | ResNet-18 | 47.73 | 0.4510 | 24.86 |
| | ViT-base | **61.88** | **0.3979** | **23.03** |

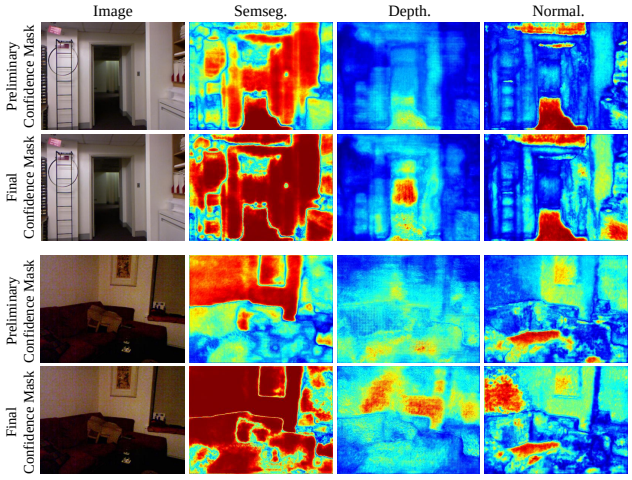


**Figure 10: Comparison of the task confidence map before and after refined by the fine-grained task tokens, which greatly encourage high-confidence predictions on all tasks (red color represents high-confidence areas). For noisy data like the second photo taken in a dark environment, this enhancement are more significant.**

**Table 13: Ablation study on the hyperparameter $\tau_i$.**

| $\tau_i$ | Semseg. *mIoU*↑ | Depth. *mErr*↓ | Normal. *mErr*↓ | $\Delta_{MTL}$ (%)↑ |
|---|---|---|---|---|
| 0.95 | 46.67 | 0.4585 | 25.48 | 3.11 |
| 0.90 | 47.10 | 0.4573 | 25.43 | 3.57 |
| 0.70 | 46.89 | 0.4621 | 25.36 | 3.17 |
| 0.50 | 47.00 | 0.4690 | 25.35 | 2.79 |
| 0.30 | 47.11 | 0.4706 | 25.33 | 2.78 |

**Table 14: Comparisons with incorporating general semi-supervised dense prediction method [43] on NYUD-v2 *one-label* setting.**

| Method | Semseg. *mIoU*↑ | Depth. *mErr*↓ | Normal. *mErr*↓ |
|---|---|---|---|
| Ours | 47.30 | 0.4539 | 25.40 |
| Ours+[43] | 47.72 | 0.4502 | 25.33 |

HiTTs with a Vision Transformer (ViT [15]) backbone. The results on NYUD-v2 are presented in Table 16. The quantitative results clearly show that when equipped with the more powerful ViT-base backbone, our method achieves a substantial performance improvement across all tasks under both the *one-label* and *random-labels* settings. For instance, in the *random-labels* setting, using ViT-base boosts the Semantic Segmentation mIoU from 47.73 to 61.88. This not only confirms the flexibility of our approach but also highlights its potential to achieve even greater performance when paired with more advanced backbone architectures.

**Analysis of Incorporating Dense FixMatch [43] on NYUD-v2.** Our method can also incorporate general semi-supervised dense prediction strategies, e.g. Dense FixMatch [43]. The quantitative results on the NYUD-v2 *one-label* setting are shown in Table 14. The results indicate that Dense FixMatch provides consistent performance improvements. This demonstrates the efficacy of Dense FixMatch as a versatile component for enhancing various dense prediction tasks in a semi-supervised context.

## B.6 More Qualitative Results

We provide more qualitative results mainly from four parts: more visualization of the fine-grained token distributions, more comparisons of task score maps produced by HiTTs, more qualitative prediction comparisons, and the quality of generated pseudo labels.

**Role of Fine-grained Task Tokens.** To illustrate the role of our fine-grained task tokens, we visualize task confidence maps before and after the refinement process in Fig. 10. The fine-grained tokens significantly enhance prediction confidence across all tasks, as indicated by the expansion of high-confidence areas (represented in red). This enhancement is particularly evident in challenging scenarios, such as the noisy image captured in a dark environment (second row), where the refined map shows a marked improvement in clarity and confidence.

**Visualization of token distributions.** We also provide visualization analysis to show the distributions of fine-grained task tokens

on Cityscapes. In Fig. 11, with the aid of OE, the self-correlation map of tokens will be more diagonal, and the distributions after PCA have better clusters in 3-dimensional feature space.

**Comparisons of Task Score Maps.** We visualize score maps produced by global task tokens and fine-grained task tokens respectively on each task. As shown in Fig. 12, we conduct visualization on both NYUD-v2 and Cityscapes datasets. The score maps indicate the response of task features to task tokens, and the response patterns of feature maps on different tasks are very different, e.g. Semseg. features highlight areas with distinguish semantics, Depth. features focus on areas with a certain depth range and Normal. features focus on surfaces with the same orientation.

Comparing the score maps produced by tokens from different hierarchies, we find that score maps produced by global task tokens are relatively rough and noisy, while those generated by fine-grained task tokens have finer granularity and less noise, which shows the hierarchy of the HiTTs learning process. Also, we observed that the high-light areas of global task tokens are monotonous, while fine-grained task tokens can highlight more details. This phenomenon is clearly observed in Cityscapes, since the ground truths of this dataset follow the long-tail distribution, thus the global task-tokens tend to learn the category with more pixel samples, and consequently always highlight the road area as shown in Fig. 12. However, the fine-grained tokens can give attention to more details, including the vehicles and pedestrians with fewer pixel samples. Thus, it is necessary to design a hierarchical structure for tokens to learn representations with different granularity.

**Qualitative prediction comparisons with SOTA works.** We additionally provide comparisons with SOTA works on NYUD-v2 and Cityscapes. As shown in Fig. 13, we compare with MTL *baseline*, XTC [28] on NYUD-v2 three tasks, and with MTL *baseline*, MTAN [33], XTC [28] on Cityscapes two tasks. Our method shows clearly better performance in semantic understanding and accurate geometry estimations (including depth and normal estimation), indicating the effectiveness of our method.

**Visualization of Pseudo Labels.** In Fig. 14, we show pseudo task label maps generated by fine-grained task tokens. The pseudo label on different tasks has good quality without ground-truth supervision, which proves the effective cross-task learning and strong generalization ability brought by HiTTs.

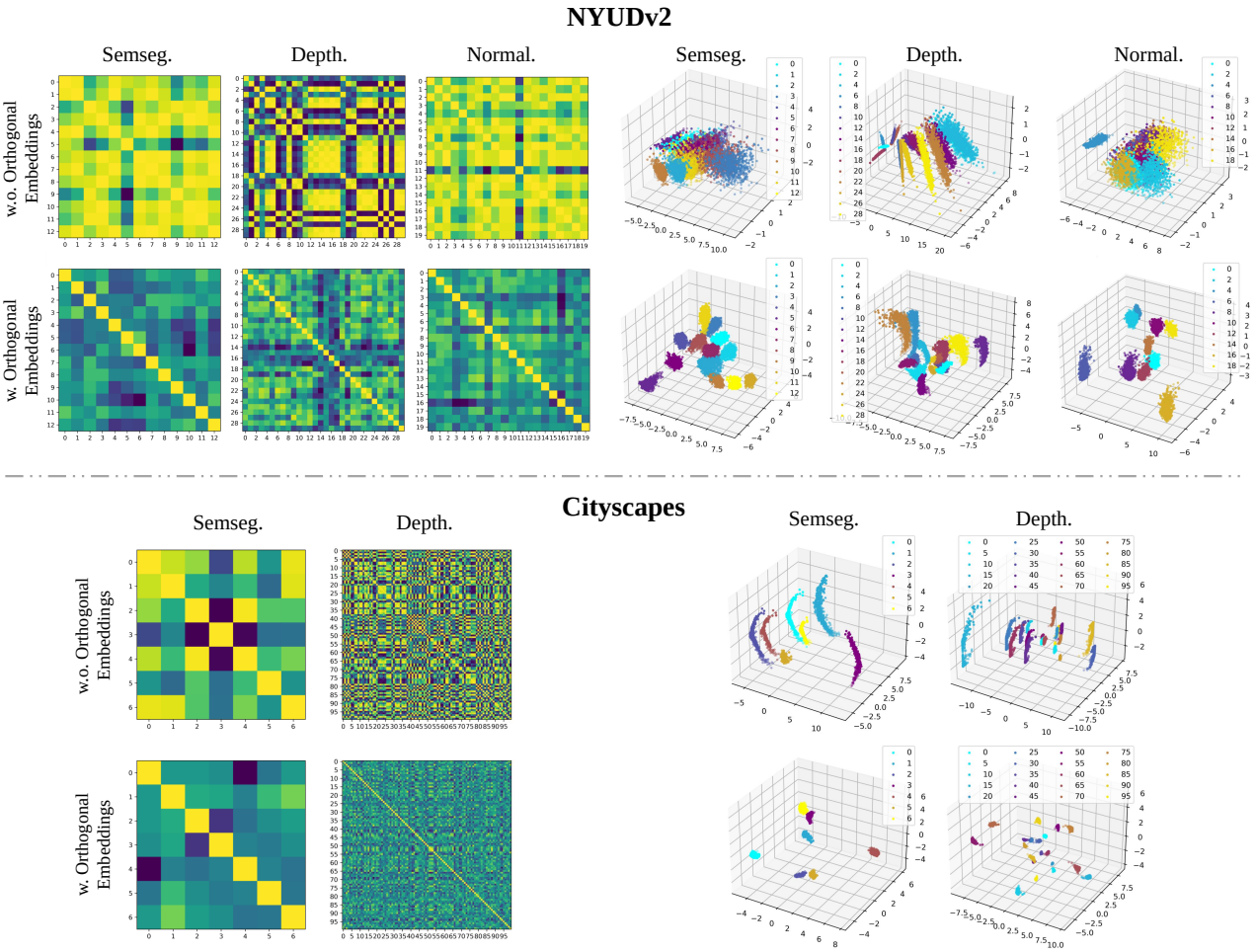


**Figure 11: Visualization of self-affinities heatmap (*left*) and PCA for distributions (*right*) of fine-grained task tokens of the two tasks on NYUDv2 and Cityscapes validation sets. With orthogonal embeddings, the affinities between different tokens are low and the clustering of token distributions on each category is better.**

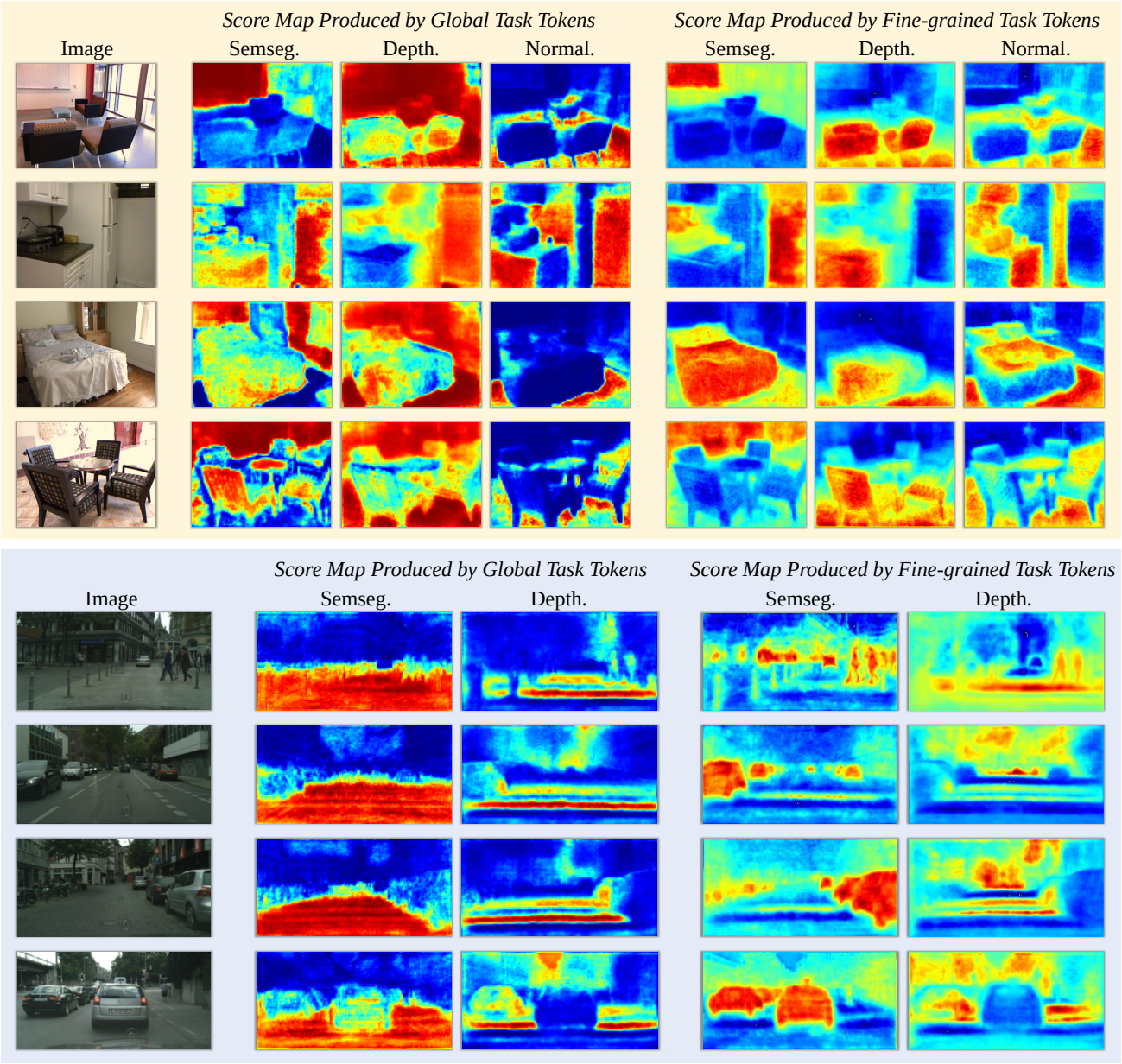


**Figure 12: Comparisons of task score maps produced by global task tokens and fine-grained task tokens. The upper part is the visualization of samples on NYUD-v2 while the lower part is on Cityscapes.**

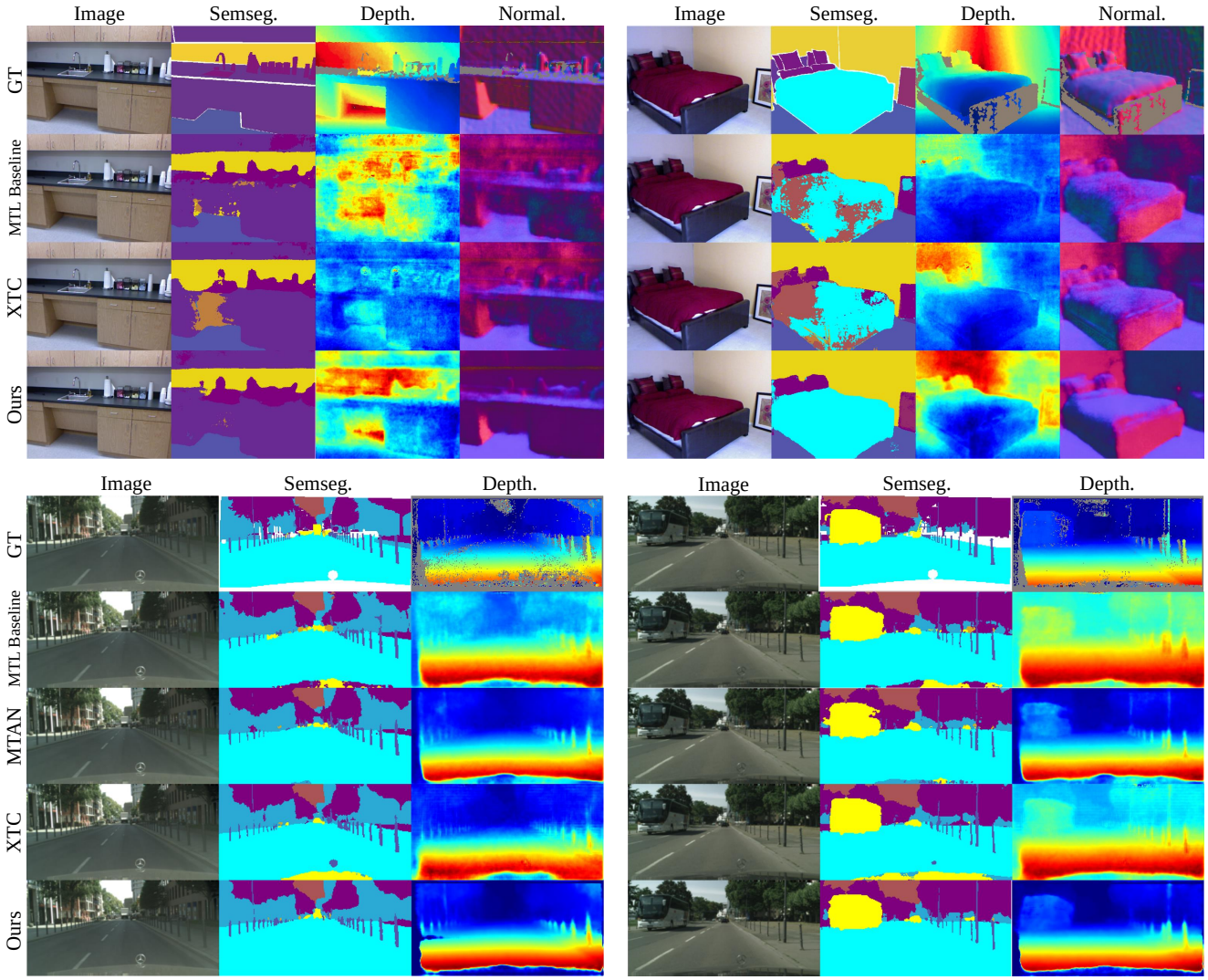


**Figure 13: Comparisons with SOTA works on NYUD-v2 (upper part) and Cityscapes (lower part).**

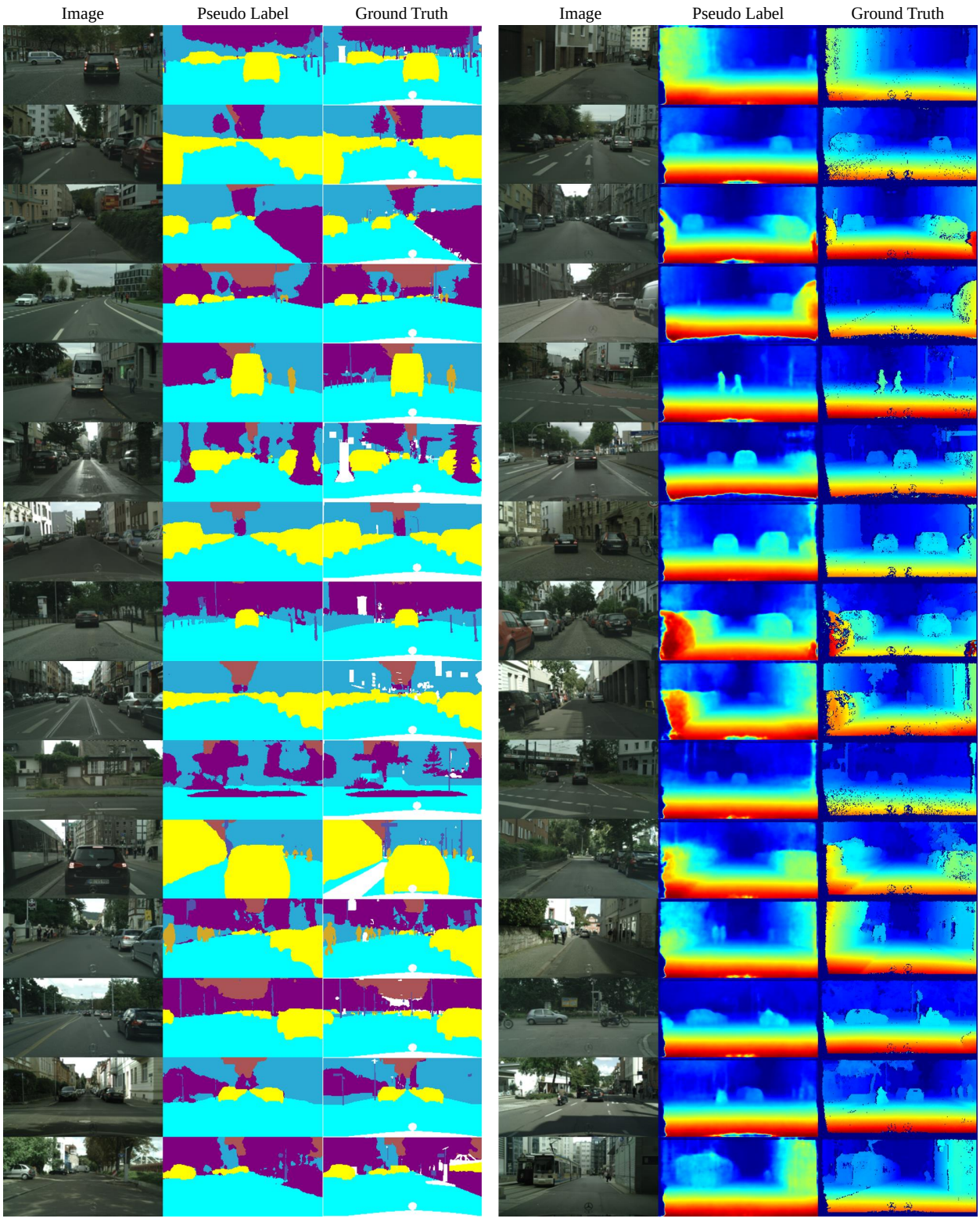


**Figure 14: Quantitative analysis of the quality of pseudo labels generated by gloabl task tokens.**