DiffusionMTL: Learning Multi-Task Denoising Diffusion Model from Partially Annotated Data

Hanrong Ye and Dan Xu[✉]
Department of Computer Science and Engineering, HKUST
Clear Water Bay, Kowloon, Hong Kong
{hyeae, danxu}@cse.ust.hk

Abstract

Recently, there has been an increased interest in the practical problem of learning multiple dense scene understanding tasks from partially annotated data, where each training sample is only labeled for a subset of the tasks. The missing of task labels in training leads to low-quality and noisy predictions, as can be observed from state-of-the-art methods. To tackle this issue, we reformulate the partially-labeled multi-task dense prediction as a pixel-level denoising problem, and propose a novel multi-task denoising diffusion framework coined as DiffusionMTL. It designs a joint diffusion and denoising paradigm to model a potential noisy distribution in the task prediction or feature maps and generate rectified outputs for different tasks. To exploit multi-task consistency in denoising, we further introduce a Multi-Task Conditioning strategy, which can implicitly utilize the complementary nature of the tasks to help learn the unlabeled tasks, leading to an improvement in the denoising performance of the different tasks. Extensive quantitative and qualitative experiments demonstrate that the proposed multi-task denoising diffusion model can significantly improve multi-task prediction maps, and outperform the state-of-the-art methods on three challenging multi-task benchmarks, under two different partial-labeling evaluation settings. The code is available at https://prismformore.github.io/diffusionmtl/.

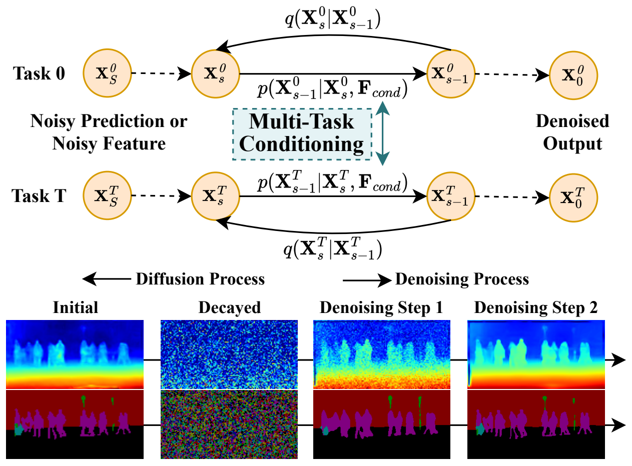

Figure 1. Motivative illustration of the proposed DiffusionMTL for multi-task partially supervised dense prediction. The model denoise the manually decayed multi-task prediction or feature maps (denoted as $\{\mathbf{X}_S^0, ..., \mathbf{X}_S^T\}$, T, S are the numbers of tasks and steps separately) in a step by step manner, and obtain the denoised outputs $\{\mathbf{X}_0^0, ..., \mathbf{X}_0^T\}$. The denoising process is guided by the designed multi-task condition feature $\mathbf{F}_{cond}$.

1. Introduction

Multi-task learning for dense scene understanding [1–4] is an important research topic that has recently gained a lot of attention from computer vision researchers. It aims at jointly learning multiple scene-related dense prediction tasks, including semantic segmentation, surface normal estimation, depth estimation, etc. This multi-task learning problem has dual superiority over traditional single-task learning. On the one hand, multi-task models are naturally more efficient than single-task models with similar structures because different tasks can share some network modules. On the other hand, different tasks are able to help each other and improve overall performance by sharing information through cross-task consistency [5]. However, annotating a real-world multi-task learning dataset at the pixel level is a daunting task. As an alternative, collecting data annotated for different tasks and using them to train a multi-task model is a much more feasible approach. This motivates recent work [6] that defines an important new problem known as “Multi-Task Partially Supervised Learning (MTPSL)”, where each training sample contains labels for a subset of the tasks, rather than all tasks. As there is a lack of multi-task labels for each training sample, the partially supervised multi-task learning problem is more challenging compared to the fully supervised multi-task learning problem. To handle this problem, previous state-of-the-art models [6] focus on improving label efficiency by enforcing cross-task consistency. They train an additional network to construct a joint feature space for each task pair, which helps improve the multi-task optimization process and demonstrates promising multi-task performance under MTPSL. Despite their success in improving model performance, the sparsity of training labels in MTPSL still inevitably leads to noisy prediction maps which can be observed from previous state-of-the-art models, as shown in Figure 7. Therefore, there is a need for a new methodology to effectively denoise the noisy multi-task dense predictions to improve the multi-task prediction quality.

To address the above-mentioned problem, we propose a novel multi-task denoising diffusion framework that can effectively remove noise from the dense predictions and rectify multi-task prediction maps. We formulate the multi-task dense prediction problem as a joint pixel-level denoising and generation process, and propose a new multi-task model coined as “DiffusionMTL”. DiffusionMTL learns to denoise noisy multi-task predictions with the help of diffusion models [7], which are particularly effective in recovering data distribution from noisy input. It jointly performs the diffusion and denoising processes to discover potential noisy distributions of the multi-task prediction maps, and learns to rectify the prediction maps. We further present two distinct diffusion mechanisms: Prediction Diffusion and Feature Diffusion. Prediction Diffusion learns to remove noise from the multi-task prediction maps, while Feature Diffusion learns to refine the multi-task feature maps. Unlike typical diffusion models used for image synthesis [8], our denoising network is designed to achieve two objectives simultaneously. Firstly, it must reverse the Markovian noise diffusion process, *i.e*., remove the manually added noise from the input maps. Secondly, it is encouraged to generate higher-quality multi-task predictions from the noisy input, thereby improving overall multi-task prediction performance. Furthermore, to exploit multi-task consistency in the denoising process, we design a Multi-Task Conditioning mechanism for our DiffusionMTL model. This mechanism utilizes the prediction maps generated by the decoders of all the tasks to effectively facilitate the denoising process of a target task. Meanwhile, the outputs of the unlabeled tasks also receive supervision signals from the ground-truth labels of other tasks, allowing our DiffusionMTL to not only enhance the denoising performance of the labeled tasks but also facilitate the learning of unlabeled tasks.

To evaluate the effectiveness of our approach for multi-task partially supervised learning, we have conducted extensive experiments on three challenging partially-annotated multi-task datasets, namely PASCAL, NYUD, and Cityscapes. Both quantitative and qualitative results demonstrate the effectiveness of the proposed Diffusion-MTL model, and show that DiffusionMTL significantly outperforms the current state-of-the-art method by a large margin, using the same backbone and fewer model parameters.

In summary, the contribution of this paper is threefold:

- We propose the first multi-task denoising diffusion framework for the partially labeled multi-task dense prediction problem. The innovative framework reformulates multi-task dense prediction as a joint pixel-level diffusion and denoising process, which empowers us to generate rectified higher-quality multi-task predictions.
- We develop a novel Multi-Task Denoising Diffusion Network specifically designed to address the issue of noise in initial prediction maps. An effective Multi-Task Conditioning mechanism is designed in our diffusion model to enhance the denoising performance. We further devise two effective diffusion mechanisms, namely Prediction Diffusion and Feature Diffusion, for refining task signals in prediction and feature spaces separately.
- Extensive experiments have been conducted on three prevalent partial-labeling multi-task benchmarks under two different settings, which clearly validate the effectiveness of our proposal. Our method demonstrates significant performance improvements compared to the previous state-of-the-art methods.

2. Related Work

Multi-Task Dense Scene Understanding with Partially Annotated Data Multi-task learning (MTL) for dense scene understanding has been widely studied in recent years [1, 3, 5, 9–16]. By learning several tasks together, MTL enhances the computational efficiency of both training and inference compared to single-task models while achieving better performance [4, 17, 18]. To improve the performance of multi-task learning, some researchers have focused on improving the optimization process of MTL by designing loss functions [2, 6, 19–21] and manipulating gradients [22–26], while other researchers work on designing powerful multi-task model architectures [27–37]. It is worth noting that the aforementioned methods are mainly designed for fully-supervised multi-task learning, where the labels of all tasks are assumed to be given for each training image. However, in real-world scenarios, it is not always feasible to obtain labels for all tasks, and we may have data with only some of the tasks available. To address this issue, a new problem named Multi-Task Partially Supervised Learning (MTPSL) has been defined by [6]. In MTPSL, the training samples are only partially annotated for the tasks, which poses new challenges to multi-task learning due to the sparsity of labels in the training data. To meet the challenge, XTC [6] has been proposed to better leverage partial annotations by improving label efficiency. It maps the label spaces of different tasks into one joint feature space and utilizes cross-task consistency to learn tasks without labels for each training sample. Although this approach has shown

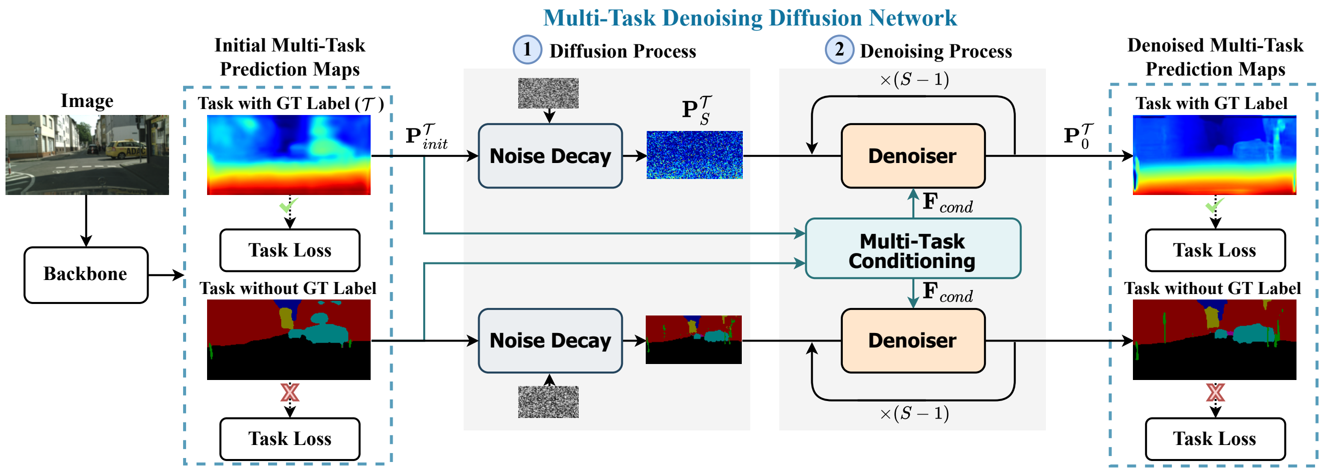

Figure 2. Illustration of the proposed DiffusionMTL (Prediction Diffusion) framework for the MTPSL setting. DiffusionMTL first uses an initial backbone model for producing starter prediction maps for all tasks. To denoise the initial prediction maps and generate rectified maps, we propose a **Multi-Task Denoising Diffusion Network** (MTDNet). MTDNet involves a diffusion process and a denoising process. During training, the initial prediction map of the labeled target task $\mathcal{T}$ is gradually degraded by applying noise, resulting in the noisy prediction map $\mathbf{P}^{\mathcal{T}}_S$. Then, we utilize a Multi-Task Conditioned Denoiser (referred to as the "Denoiser") to denoise $\mathbf{P}^{\mathcal{T}}_S$ iteratively over S steps, resulting in a clean prediction map $\mathbf{P}^{\mathcal{T}}_0$ that is supervised by the ground-truth label. For better learning of unlabeled tasks, we propose a **Multi-Task Conditioning** mechanism in the denoising process to stimulate information sharing across different tasks. During inference, the diffusion and denoising processes are applied to all tasks to produce denoised multi-task prediction maps.

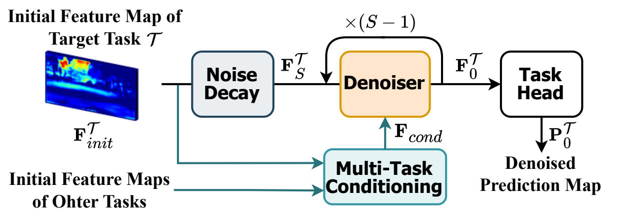
Initial Feature Map of Target Task $\mathcal{T}$
$\mathbf{F}^{\mathcal{T}}_{init}$
Initial Feature Maps of Ohter Tasks
Noise Decay
$\mathbf{F}^{\mathcal{T}}_{S}$
$\times(S-1)$
Denoiser
$\mathbf{F}_{cond}$
Multi-Task Conditioning
$\mathbf{F}^{\mathcal{T}}_{0}$
Task Head
$\mathbf{P}^{\mathcal{T}}_{0}$
Denoised Prediction Map

Figure 3. Illustration of the proposed DiffusionMTL (Feature Diffusion), which conducts noise decay and denoising on initial feature maps $\mathbf{F}^{\mathcal{T}}_{init}$. The denoised feature maps $\mathbf{F}^{\mathcal{T}}_0$ are projected to the final prediction map $\mathbf{P}^{\mathcal{T}}_0$ with a task head after the denoising.

promising results, it still inevitably suffers from noisy predictions because the model is under-trained with a limited number of ground-truth labels. To directly tackle the noisy prediction problem, our proposal takes a distinct approach by designing a novel multi-task denoising framework to improve the quality of multi-task prediction maps.

Diffusion Models Diffusion models [7, 38] are a class of generative models that have been widely used for image synthesis tasks, and have achieved state-of-the-art performance on several benchmarks [7, 8, 39–41]. However, adapting diffusion models for multi-task dense prediction is not straightforward. Although some attempts have been made to apply diffusion models to single-task deterministic problems including image classification [42], segmentation [43–47] and detection [48], they are not suitable for multi-task dense scene understanding. In this paper, we propose a novel multi-task diffusion model that can denoise noisy prediction maps for multiple tasks and obtain finer results under the multi-task partially-supervised setting. Our approach represents an exploratory advancement in diffusion models and has the potential to inspire the design of future diffusion models for deterministic tasks.

3. The Proposed DiffusionMTL Approach

In this section, we will introduce the details of our proposed multi-task denoising diffusion framework, DiffusionMTL, as illustrated in Fig. 2. DiffusionMTL has two steps: (i) First, an initial backbone model generates preliminary prediction maps for multiple dense scene understanding tasks. (ii) Second, a proposed Multi-Task Denoising Diffusion Network (MTDNet) takes in the noisy initial multi-task prediction maps and produces refined prediction results. These two parts are trained together in an end-to-end manner with partially annotated data.

3.1. Initial Backbone Model

We adopt a classic encoder-decoder structure for the multi-task dense prediction [4, 17, 32]. The initial backbone model utilizes a task-shared encoder f_{enc}, which accepts an input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ (where H and W represent height and width, respectively) and projects it to obtain a multi-channel backbone feature map $\mathbf{F}_{backbone} \in \mathbb{R}^{H'\times W'\times C}$. The backbone feature map has a height of H', a width of W', and C channels. It is shared by all the tasks. And then, to generate task-specific feature maps for T tasks, we adopt a series of task-specific decoders $\{f^1_{dec}, f^2_{dec}, ..., f^T_{dec}\}$ with identical network structures and different network parameters. The generated initial task feature maps from the decoders are notated as $\{\mathbf{F}^1_{init}, \mathbf{F}^2_{init}, ..., \mathbf{F}^T_{init}\}$. For the t-th task, we compute the task-specific initial feature map $\mathbf{F}^t_{init}$ as:

$$\mathbf{F}^t_{init} = f^t_{dec}(f_{enc}(\mathbf{I})). \quad (1)$$

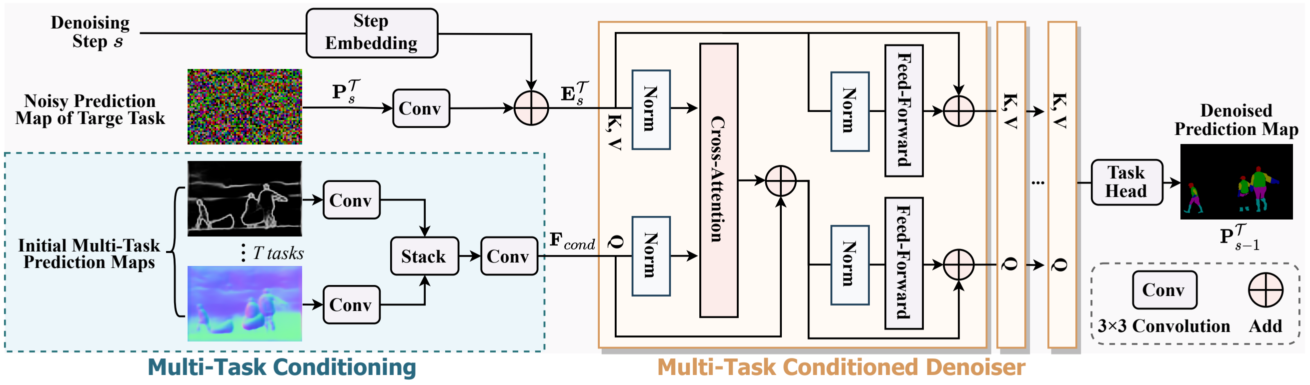
Denoising Step s
Step Embedding
Noisy Prediction Map of Targe Task
$\mathbf{P}_s^{\mathcal{T}}$
Conv
$\mathbf{E}_s^{\mathcal{T}}$
Initial Multi-Task Prediction Maps
T tasks
Conv
Stack
Conv
$\mathbf{F}_{cond}$
K, V
Q
Norm
Cross-Attention
Feed-Forward
Task Head
Denoised Prediction Map
$\mathbf{P}_{s-1}^{\mathcal{T}}$
3×3 Convolution
Add
Multi-Task Conditioning
Multi-Task Conditioned Denoiser

Figure 4. Pipeline of a single step s in the denoising process of DiffusionMTL (Prediction Diffusion). **Multi-Task Conditioning**: The initial prediction maps for all tasks are projected to task-specific features and then stacked. The stacked features are then processed with a 3×3 convolution to reduce the channel dimension, resulting in a Multi-Task Condition Feature $\mathbf{F}_{cond}$ which is shared across all tasks. **Multi-Task Conditioned Denoiser**: The denoiser consists of several cross-attention transformer blocks, which learn to denoise input conditioned on $\mathbf{F}_{cond}$. For its input, we perform a 3×3 convolution on the noisy prediction map $\mathbf{P}_s^{\mathcal{T}}$ and combine the output with the step embedding, obtaining a task embedding $\mathbf{E}_s^{\mathcal{T}}$. The denoiser takes $\mathbf{F}_{cond}$ as query input and $\mathbf{E}_s$ as key and value inputs. We use a task-specific head to obtain the denoised prediction map $\mathbf{P}_{s-1}^{\mathcal{T}}$, which is the input of the next denoising step $s-1$.

To compute an initial dense prediction map $\mathbf{P}_{init}^t$ for the t-th task, we apply a task-specific 1×1 convolution f_{pred}^t on the corresponding task feature map $\mathbf{F}_{init}^t$:

$$\mathbf{P}_{init}^t = f_{pred}^t(\mathbf{F}_{init}^t). \tag{2}$$

In this way, we obtain the T initial prediction maps of all T tasks. The initial prediction maps are noisy as we can observe from Fig. 2. We aim to rectify the noisy multi-task prediction maps with the following MTDNet.

3.2. Multi-Task Denoising Diffusion Network

In this paper, we put forward a novel diffusion model, named Multi-Task Denoising Diffusion Network (MTDNet) for denoising the aforementioned noisy prediction maps. To achieve this goal, we design two orthogonal diffusion mechanisms in our unified MTDNet, focusing on different signal domains: **(i) Prediction Diffusion and (ii) Feature Diffusion**. These mechanisms differ in terms of the signal space in which the diffusion model is applied. Feature Diffusion refines the task-specific features within a high-dimensional latent space, while Prediction Diffusion directly improves the initial task predictions in the output space. Feature Diffusion facilitates a comprehensive improvement of high-level visual information within an expanded latent space, while Prediction Diffusion demonstrates effective denoising capabilities along with better computational efficiency. More details about their difference will be provided when describing the different components of MTDNet. As shown in Fig. 2 for Prediction Diffusion and Fig. 3 for Feature Diffusion, we follow the DDPM paradigm [7], which is separated into 2 processes: a diffusion process and a denoising process. During the diffusion process, we incrementally degrade the information in the initial prediction maps by applying noise with a Markovian chain. In the denoising process, we introduce a novel denoising network that is trained to generate clean prediction maps from the degraded ones in an iterative manner. Without loss of generality, we elaborate on the one-label setting of the multi-task partially supervised learning, where we assume that the current training sample has label for only one task (task $\mathcal{T}$). We name this labeled task as the "target task".

3.2.1 Diffusion Process

In the diffusion process (or "forward process") [7], we construct a fixed Markov chain with a total length of S steps. For the target task $\mathcal{T}$, Gaussian noise is gradually applied to the initial map $\mathbf{X}_{init}^{\mathcal{T}}$ in a step-by-step manner. Here $\mathbf{X}_{init}^{\mathcal{T}}$ is the initial prediction map $\mathbf{P}_{init}^{\mathcal{T}}$ (in Prediction Diffusion) or initial task feature map $\mathbf{F}_{init}^{\mathcal{T}}$ (in Feature Diffusion). Suppose the decayed map at denoising step s of target task $\mathcal{T}$ is $\mathbf{X}_s^{\mathcal{T}}$, the diffusion process q can be formulated as:

$$q(\mathbf{X}_s^{\mathcal{T}}|\mathbf{X}_{init}^{\mathcal{T}}) = \mathcal{N}(\mathbf{X}_s^{\mathcal{T}}|\sqrt{\bar{\alpha}_s}\mathbf{X}_{init}^{\mathcal{T}}, (1-\bar{\alpha}_s)\mathbf{I}), \tag{3}$$

where $\{\bar{\alpha}_s, s \in 1, 2, ..., S\}$ are hyperparameters. In practice, we could directly compute the final decayed map. Through mathematical derivation, the decayed prediction map of target task $\mathcal{T}$ at final step S can be formulated as $\mathbf{X}_S^{\mathcal{T}} = \sqrt{\bar{\alpha}_S}\mathbf{X}_{init}^{\mathcal{T}} + \sqrt{1-\bar{\alpha}_S}\epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. For more theoretical details please refer to [7].

3.2.2 Denoising Process

As the core component of our MTDNet, the denoising process involves designing a Multi-Task Conditioned Denoiser, referred to as "Denoiser", to denoise the noisy multi-task prediction maps or feature maps. Specifically, given the decayed noisy map of target task $\mathbf{X}_S^{\mathcal{T}}$ from the diffusion

Pseudocode 1 DiffusionMTL under one-label setting

```
1: function DiffusionMTL(version, mode, $\mathbf{I}$, $\mathbf{L}$, $\mathcal{T}$, $S$, $T$)
2:     Input: version ∈ {'Prediction Diffusion', 'Feature Diffusion'}, mode ∈ {'train', 'infer'}, input image $\mathbf{I}$, label map $\mathbf{L}$, target task $\mathcal{T}$, diffusion steps $S$, number of tasks $T$
3:     Output: Training loss or denoised prediction map $\mathbf{P}_0^{\mathcal{T}}$
4:     $\mathbf{F}_{backbone} \leftarrow f_{enc}(\mathbf{I})$
5:     for $t \leftarrow 1, 2, \ldots, T$ do
6:         $\mathbf{F}_{init}^t \leftarrow f_{dec}^t(\mathbf{F}_{backbone})$        ▷ Compute initial task features
7:         $\mathbf{P}_{init}^t \leftarrow f_{pred}^t(\mathbf{F}_{dec}^t)$        ▷ Compute initial prediction maps
8:     end for
9:     if version = 'Prediction Diffusion' then
10:        $\mathbf{X}_{init}^{\mathcal{T}} \leftarrow \mathbf{P}_{init}^{\mathcal{T}}$
11:        $\mathbf{F}_{cond} \leftarrow f_{cond}(\mathbf{P}_{init}^1, \mathbf{P}_{init}^2, \ldots, \mathbf{P}_{init}^T)$
12:    else if version = 'Feature Diffusion' then
13:        $\mathbf{X}_{init}^{\mathcal{T}} \leftarrow \mathbf{F}_{init}^{\mathcal{T}}$
14:        $\mathbf{F}_{cond} \leftarrow f_{cond}(\mathbf{F}_{init}^1, \mathbf{F}_{init}^2, \ldots, \mathbf{F}_{init}^T)$
15:    end if
16:    $\epsilon \sim \mathcal{N}(0, \mathbf{I})$        ▷ Sample noise
17:    $\mathbf{X}_S^{\mathcal{T}} \leftarrow \sqrt{\bar{\alpha}_S}\mathbf{X}_{init}^{\mathcal{T}} + \sqrt{1-\bar{\alpha}_S}\epsilon$        ▷ Diffusion process
18:    for $s \leftarrow S, S-1, \ldots, 1$ do
19:        $\mathbf{X}_{s-1}^{\mathcal{T}} \leftarrow$ Denoiser($\mathbf{X}_s^{\mathcal{T}}, s, \mathbf{F}_{cond}$)        ▷ Denoising step
20:    end for
21:    if version = 'Prediction Diffusion' then
22:        $\mathbf{P}_0^{\mathcal{T}} \leftarrow \mathbf{X}_0^{\mathcal{T}}$
23:    else if version = 'Feature Diffusion' then
24:        $\mathbf{P}_0^{\mathcal{T}} \leftarrow f_{head}^{\mathcal{T}}(\mathbf{X}_0^{\mathcal{T}})$        ▷ Final task head
25:    end if
26:    if mode = 'train' then
27:        return compute_loss($\mathbf{P}_0^{\mathcal{T}}, \mathbf{L}$)        ▷ Compute loss with available label of target task
28:    else if mode = 'infer' then
29:        return $\mathbf{P}_0^{\mathcal{T}}$        ▷ Output denoised prediction map
30:    end if
31: end function
```

process, Denoiser generates $\mathbf{X}_{S-1}^{\mathcal{T}}$, $\mathbf{X}_{S-2}^{\mathcal{T}}$,..., $\mathbf{X}_0^{\mathcal{T}}$ in an iterative manner. In the following, we will first introduce a novel Multi-Task Conditioning strategy, and then describe how Denoiser computes the denoised map in each denoising step. Fig. 4 illustrates the computation pipeline for a single denoising step of Prediction Diffusion.

Multi-Task Conditioning To help denoise the prediction or feature maps of the labeled tasks, as well as enable learning unlabeled tasks under a partially annotated setting, we propose a Multi-Task Conditioning strategy in the denoising process. It first obtains a "multi-task condition feature" (denoted as $\mathbf{F}_{cond}$) from the initial maps of all the tasks. $\mathbf{F}_{cond}$ captures the joint multi-task information, which is later used to condition the denoising network. To obtain $\mathbf{F}_{cond}$, we first project the initial multi-task maps to feature space via a 3×3 convolution and obtain task-specific features for all T tasks. Once we have obtained task-specific features for all T tasks, we combine them along the channel dimension. The resulting tensor is then subjected to a 3×3 convolutional layer, which reduces the channel dimension to C and then flattens the spatial dimension, resulting in a vector known as the multi-task condition feature $\mathbf{F}_{cond}$. We refer to this computational process as f_{cond}.

Multi-Task Conditioned Denoiser We illustrate the structure of denoiser in Fig. 4. The denoiser is a series of cross-attention transformer blocks, taking in the noisy map $\mathbf{X}_s^{\mathcal{T}}$, the denoising step s, and the multi-task condition feature $\mathbf{F}_{cond}$ as input, and generates the denoised map $\mathbf{X}_{s-1}^{\mathcal{T}}$. $\mathbf{X}_{s-1}^{\mathcal{T}}$ is used as input for the next step in an iterative manner. Specifically, we start by projecting the noisy map of the target task to a C-channel task embedding via a 3×3 convolution and flatten its spatial dimension. Then we embed the denoising step s using a typical sinusoidal embedding module [8]. We call this process "Step Embedding". We add the step embedding to the task embedding. The resulting task embedding $\mathbf{E}_s^{\mathcal{T}}$ assumes the role of key and value tensors $(\mathbf{K}, \mathbf{V})$ in the subsequent transformer blocks, and $\mathbf{F}_{cond}$ is supplied to the transformer blocks as the query $\mathbf{Q}$:

$$\mathbf{Q} \leftarrow \mathbf{F}_{cond}, \ \ \mathbf{K} \leftarrow \mathbf{E}_s^{\mathcal{T}}, \ \ \mathbf{V} \leftarrow \mathbf{E}_s^{\mathcal{T}}. \tag{4}$$

The transformer blocks receive $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ as input. Each block comprises linear normalization, cross-attention, and feed-forward networks as shown in Fig. 4. For a more comprehensive understanding of the details of transformer, please consult [49]. Here, the cross-attention transformer blocks absorb information from the task embedding $\mathbf{E}_s^{\mathcal{T}}$ guided by the multi-task conditioning feature $\mathbf{F}_{cond}$.

The output procedure of a denoising step is different in Prediction Diffusion and Feature Diffusion. In Prediction Diffusion, the output of the transformer blocks is reshaped to a spatial map and projected to prediction map $\mathbf{P}_{s-1}^{\mathcal{T}}$ using a task-specific head which consists of several convolutional layers with ReLU. In Feature Diffusion, the output of transformer blocks is projected to a feature map $\mathbf{F}_{s-1}^{\mathcal{T}}$, which serves as the output of this step (*i.e.* $\mathbf{X}_{s-1}^{\mathcal{T}}$). More implementation details can be found in Sec. 4.1. We can formulate each step in the denoising process as:

$$\begin{aligned} \mathbf{X}_{s-1}^{\mathcal{T}} &= \text{Denoiser}(\mathbf{X}_s^{\mathcal{T}}, s, \mathbf{F}_{cond}), \\ s &\in S, S-1, ..., 1. \end{aligned} \tag{5}$$

For the final output after S denoising steps, Prediction Diffusion directly generates the denoised prediction map of target task $\mathbf{P}_0^{\mathcal{T}}$, which is used to compute task-specific loss supervised by the available ground-truth label. In Feature Diffusion, we need a final task-specific head $f_{head}^{\mathcal{T}}$ to project the denoised feature maps $\mathbf{F}_0^{\mathcal{T}}$ to the final prediction map $\mathbf{P}_0^{\mathcal{T}}$. We present the detailed training and inference pipelines of DiffusionMTL in Pseudocode 1.

3.3. Model Optimization

The whole DiffusionMTL model can be trained under the MTPSL setting in an end-to-end manner. Specifically, for each training sample, we apply task-specific losses on both the initial prediction maps as well as the final denoised prediction maps of the tasks with labels. For the unlabeled tasks, there are no ground-truth supervision signals, but the task-specific decoders are able to be implicitly trained via the proposed Multi-Task Conditioning. More details about losses are introduced in the supplemental materials.

4. Experiments

4.1. Experimental Setup

Datasets and Tasks Following the pioneering work in multi-task partially supervised learning [6], we adopt three prevalent multi-task datasets with dense annotations, *i.e.* PASCAL [50], NYUD [51], and Cityscapes [52]. **PASCAL** is a comprehensive dataset providing images of both indoor and outdoor scenes. There are 4,998 training images and 5,105 testing images, with labels of semantic segmentation, human parsing, and object boundary detection. Additionally, [1] generates pseudo labels for surface normals estimation and saliency detection. **NYUD** (or NYUD-v2) provides images of indoor scenes as well as dense annotations for 13-class semantic segmentation and depth estimation. The images are resized to 288×384. The surface normals can be generated from depth. The training set contains 795 images, while the testing set contains 654 images. **Cityscapes** captures street scenes of different cities with fine pixel-level annotations. Following [6, 10], we use 7-class semantic segmentation and monocular depth estimation tasks in the experiments. The images are resized to 128×256. There are 2,975 training images and 500 validation images.

Task Metrics We adopt the same metrics for different tasks as previous work [6]. We use the mean Intersection over Union (mIoU) to evaluate the performance of the semantic segmentation (Semseg) and human parsing (Parsing) tasks, while the absolute error (absErr) is used for evaluating the monocular depth estimation task (Depth). For the surface normal estimation task (Normal), we use the mean error of angles (mErr) as the evaluation metric, while for the saliency detection task (Saliency), we use the maximal F-measure (maxF). The object boundary detection task (Boundary) is evaluated using the optimal-dataset-scale F-measure (odsF). To quantify the overall multi-task performance relative to the single-task baseline, we calculate the mean relative difference across all tasks, denoted as Multi-task Performance (MTL Perf Δ_m) [1].

MTPSL Evaluation Settings There are two evaluation settings for multi-task partially supervised learning [6]: (i) **one-label setting**, where each training image has the ground-truth label of only one task. (ii) **random-label setting**, where the number of labeled tasks for each image is random. We use exactly the same image-task label mappings as [6] for a strictly fair comparison.

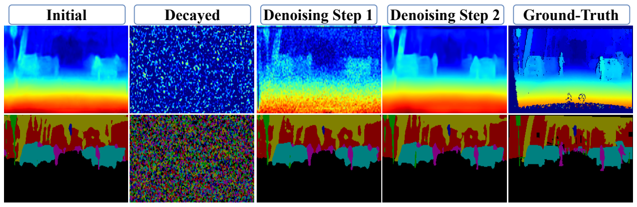

Figure 5. Visualization of the prediction maps at different processes on Cityscapes. Our DiffusionMTL effectively denoises the noisy prediction maps of both tasks.

# labels	Method	Semseg mIoU ↑	Depth absErr ↓	MTL Perf Δ_m ↑
	Single-Task	75.82	0.0125	-
one / random	MTL Baseline	73.19	0.0168	-18.81%
	SS [6]	71.67	0.0178	-
	XTC [6]	74.90	0.0161	-
	XTC* [6]	73.36	0.0158	-14.74%
	DiffusionMTL (Prediction)	74.90	0.0131	-2.79%
	DiffusionMTL (Feature)	**75.67**	**0.0130**	**-2.13%**

Table 1. Comparison with SOTAs on Cityscapes. The proposed DiffusionMTL demonstrates superior performance on both tasks. One-label setting is equivalent to the random-label setting on Cityscapes. "*" denotes re-implemented results.

Implementation Details For most models in the experiments, we use ResNet-18 as backbone with ImageNet pretrained weights provided by PyTorch. We concatenate the feature maps of the four stages of ResNet-18 along the channel dimension and process them with a 3×3 convolution to reduce the number of channels to 512. For our DiffusionMTL, the initial multi-task backbone has a task-specific decoder for each task, using 2 residual convolutional blocks [53], followed by a 1×1 convolution as prediction head. Each residual convolutional block contains two 3×3 convolutions with BN and ReLU. The denoising network uses 4 task-shared cross-attention transformer blocks, which are followed by a task-specific head of four 3×3 CONV-ReLU layers and a 1×1 convolution for task prediction in Prediction Diffusion. We use 2 steps in the diffusion process. More implementation details can be found in the supplemental materials.

4.2. Main Experiments

Declaration of Comparison Models We consider several models for comparison to verify the effectiveness of our proposed DiffusionMTL framework: (i) "MTL Baseline" is the baseline model. It shares the same backbone as DiffusionMTL and utilizes a strong task-specific decoder for each task. It consists of 6 residual convolutional blocks, followed by a 1×1 convolution as prediction head. (ii) "SS" and "XTC" [6] are pioneering state-of-the-art methods as introduced in related work. XTC is re-implemented on our MTL Baseline based on their official codes for fair com-

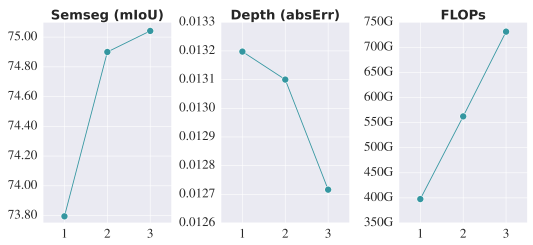

Figure 6. Study of training DiffusionMTL with varying numbers of diffusion steps on Cityscapes. Adding diffusion steps increases the performance and FLOPs.

Backbone	Method	#Params	FLOPS	Semseg mIoU ↑	Parsing mIoU ↑	Saliency maxF ↑	Normal mErr ↓	Boundary odsF ↑	MTL Perf Δ_m ↑
R18	**DiffusionMTL (Feature)**	133M	676G	57.78	**58.98**	**77.82**	**16.11**	**64.50**	**+3.65%**
	DiffusionMTL (Prediction)	133M	628G	**59.43**	56.79	77.57	16.20	64.00	+3.23%
	—w/o Diffusion	133M	628G	57.77	56.39	77.44	17.34	60.60	+0.11%
	—w/o Multi-Task Cond	125M	558G	55.95	55.90	77.22	17.87	61.50	-1.33%
R50	**DiffusionMTL (Feature)**	159M	742G	58.78	**61.91**	77.07	**16.49**	**66.20**	**0.77%**
	DiffusionMTL (Prediction)	159M	694G	**60.92**	59.94	**77.58**	17.31	63.80	-0.66%
	—w/o Diffusion	159M	694G	58.10	58.69	76.64	17.50	62.80	-2.86%
	—w/o Multi-Task Cond	150M	625G	57.29	59.37	76.90	17.74	63.70	-2.91%

Table 2. Ablation study on PASCAL. "w/o Diffusion" indicates replacing the diffusion model with an iterative refinement model using an identical network structure. "w/o Multi-Task Cond" means removing Multi-Task Conditioning.

# labels	Method	PASCAL								NYUD			
		#Params	FLOPS	Semseg mIoU ↑	Parsing mIoU ↑	Saliency maxF ↑	Normal mErr ↓	Boundary odsF ↑	MTL Perf Δ_m ↑	Semseg mIoU ↑	Depth absErr ↓	Normal mErr ↓	MTL Perf Δ_m ↑
one	Single-Task Baseline	219M	817G	50.34	59.05	77.43	16.59	64.40	-	45.28	0.4802	25.93	-
	MTL Baseline	157M	608G	49.71	56.00	74.50	16.85	62.80	-2.85%	43.92	0.5138	26.44	-3.99%
	SS [6]	-	-	45.00	54.00	61.70	16.90	62.40	-	27.52	0.6499	33.58	-
	XTC [6]	-	-	49.50	55.80	61.70	17.00	**65.10**	-	30.36	0.6088	32.08	-
	XTC* [6]	173M	608G	55.08	56.72	77.06	16.93	63.70	+0.37%	43.97	0.5140	26.30	-3.79%
	DiffusionMTL (Prediction)	133M	628G	**59.43**	56.79	77.57	16.20	64.00	+3.23%	**44.97**	0.5137	26.17	-2.86%
	DiffusionMTL (Feature)	133M	676G	57.78	**58.98**	**77.82**	**16.11**	64.50	**+3.65%**	44.47	**0.5059**	**25.84**	**-2.27%**
random	Single-Task	219M	817G	51.51	57.90	80.30	15.24	67.80	-	48.25	0.4792	24.65	-
	MTL Baseline	157M	608G	62.23	55.88	78.67	15.47	66.70	+2.44%	45.93	0.4839	25.53	-3.12%
	SS [6]	-	-	59.00	55.80	64.00	15.90	66.90	-	29.50	0.6224	33.31	-
	XTC [6]	-	-	59.00	55.60	64.00	15.90	**67.80**	-	34.26	0.5787	31.06	-
	XTC* [6]	173M	608G	62.44	55.81	78.56	15.45	66.80	+2.52%	46.03	0.4811	25.97	-3.44%
	DiffusionMTL (Prediction)	133M	628G	**63.68**	55.84	79.87	15.38	66.80	+3.44%	**47.44**	0.4803	25.26	-1.45%
	DiffusionMTL (Feature)	133M	676G	62.55	**56.84**	**80.44**	**14.85**	67.10	**+4.27%**	46.82	**0.4743**	**24.75**	**-0.77%**

Table 3. Quantitative multi-task performance comparison with the state-of-the-arts (SOTAs) on PASCAL and NYUD. All models use a ResNet-18 as the backbone. 'one' means each training image has only one labeled task, while 'random' means each training image has a random number of labeled tasks. The proposed DiffusionMTL, including both Prediction Diffusion and Feature Diffusion, achieves significantly better performance while using fewer model parameters. "*" denotes our re-implemented results.

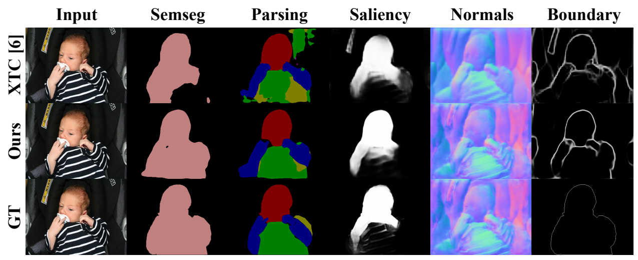

Figure 7. Qualitative comparison between the state-of-the-art XTC [6] and our method on PASCAL under one-label setting. XTC suffers from the issue of noisy predictions. In contrast, our DiffusionMTL model learns to denoise the noisy prediction maps, resulting in significantly better multi-task prediction maps.

parison. (iii) "Single-Task" is a single-task learning version of the MTL Baseline. It contains a set of separate models where each model is trained to learn only one single task.

Quantitative Comparison with SOTAs We compare the proposed DiffusionMTL with several strong competitors introduced in Sec. 4.2 on three widely-used benchmarks, *i.e*. Cityscapes, PASCAL, and NYUD. We show the results of the three benchmarks in Table 1 and Table 3, respectively. As can be observed from the tables, our DiffusionMTL demonstrates significant improvements over the competing MTL Baseline and XTC [6] on all three benchmarks. Specifically, under the challenging one-label setting on PASCAL, our DiffusionMTL (prediction) outperforms the MTL Baseline by +9.72 on Semseg and +3.07 on Saliency, while the multi-task performance Δ_m is also improved by +6.08%. Compared with the state-of-the-art method XTC, our proposal achieves an improvement of +2.86% in terms of the multi-task performance Δ_m. We can observe consistent performance gains under the random-label setting. Similarly, under the one-label setting of NYUD, our Feature Diffusion improves Δ_m by +1.52% compared with XTC. On Cityscapes, where the one-label setting is equivalent to the random-label setting, the multi-task performance Δ_m is improved by +11.95% compared with the previous best XTC. For computational efficiency, as shown in Table 3, our model consumes only 133M network parameters, which is clearly less than 173M used by XTC [6]. These significant results can fully show the effectiveness of the proposed DiffusionMTL method, which substantially outperforms the competing methods on all benchmarks while using fewer model parameters.

Qualitative Comparison with SOTAs To examine the quality of generated multi-task predictions by DiffusionMTL, we visualize the predictions by Prediction Diffusion trained under the one-label setting of PASCAL, and compare them with the outputs of SOTA model [6] as well as ground-truth labels in Fig. 7. The images are randomly chosen from the testing set of PASCAL. We observe that the generated multi-task predictions of the previous best method are noisy, which confirms our motivation to design a

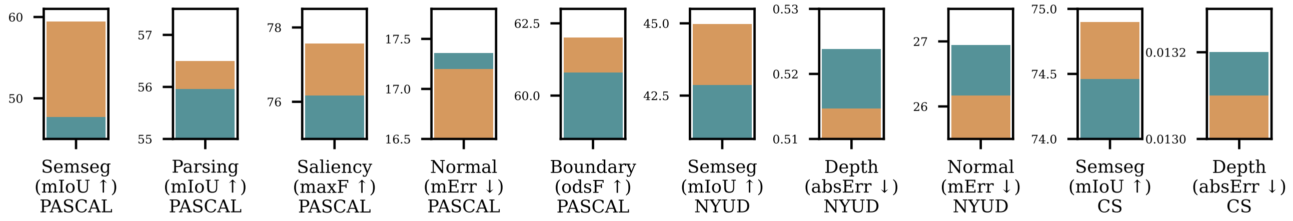

Figure 8. Performance of the initial multi-task predictions (blue) and the denoised predictions (yellow) of DiffusionMTL (Prediction) on PASCAL, NYUD, and Cityscapes under one-label setting. Our denoising network improves the prediction quality of all 10 tasks.

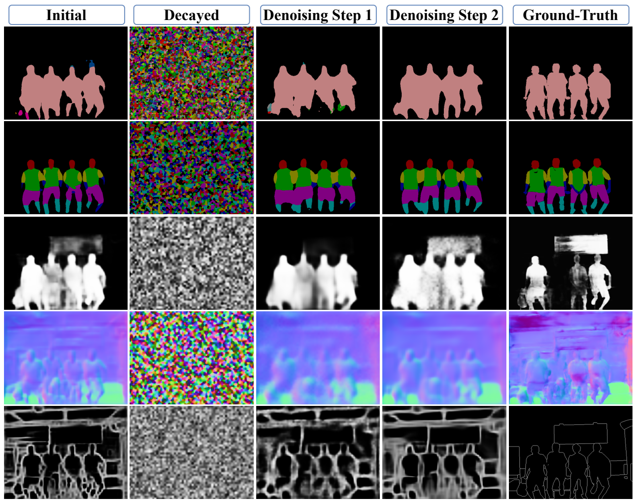

Figure 9. Visualization of the prediction maps at different processes on PASCAL. Our DiffusionMTL can denoise and rectify the noisy multi-task prediction maps.

multi-task denoising framework for the multi-task partially supervised learning problem. With the proposed DiffusionMTL, the prediction quality is significantly improved.

4.3. Ablation Study

We conduct comprehensive ablation experiments to evaluate the effectiveness of different components of DiffusionMTL for multi-task partially supervised learning and show the results on PASCAL one-label setting in Table 2.

Effectiveness of Multi-Task Denoising Diffusion Network To further confirm the effectiveness of the proposed multi-task denoising diffusion network (MTDNet), we replace it with an iterative refinement network using an identical network structure (*i.e*. cross-attention transformer blocks in Prediction Diffusion). This variant is denoted as “w/o Diffusion” in Table 2. MTDNet brings a significant multi-task performance improvement of +3.12 (ResNet-18) and +2.20 (ResNet-50) using the same computational costs, which clearly validates the effectiveness of the proposed multi-task denoising method. Moreover, we plot the performance metrics of initial predictions and final predictions in Fig. 8, which shows improvement brought by MTDNet on all 10 tasks of three benchmarks.

Effectiveness of Multi-Task Conditioning To evaluate the efficacy of the multi-task conditioning strategy, we conduct ablation experiments by removing it from DiffusionMTL (Prediction) and replacing cross-attention blocks with self-attention blocks. This variant is indicated as “w/o Multi-Task Cond”. Multi-task conditioning leads to significant improvement on all tasks, underscoring the unique importance of multi-task information sharing in the partially-labeled multi-task learning problem.

Comparison of Feature Diffusion and Prediction Diffusion As observed in Table 2, both multi-task diffusion mechanisms show significant multi-task performance on different backbones. Prediction Diffusion is more computationally efficient due to the lower dimensions of the intermediate maps, whereas Feature Diffusion achieves higher performance on most tasks by capturing more visual information in the features.

Qualitative Analysis of Denoising Effect In the denoising process, the model is designed to denoise the noisy multi-task prediction maps that are degraded by the diffusion process. To evaluate the denoising performance of our Prediction Diffusion, we provide visualizations of the multi-task prediction maps at different phases in Fig. 5 and Fig. 9. Our model generates clean and accurate multi-task prediction maps from noisy inputs, which firmly indicates the effectiveness of our multi-task denoising framework.

Influence of Diffusion Steps We plot the performance metrics against diffusion steps on Cityscapes in Fig. 6. Our observations reveal that utilizing two steps yields a notable improvement in performance compared to using just one step. Moreover, increasing the number of steps further enhances performance, albeit at a higher computational cost. Therefore, we set the default number of steps to two.

5. Conclusion

Our study aims to address the issue of noisy predictions in multi-task learning from partially annotated data. We propose a unified multi-task denoising diffusion framework that refines multi-task signals in the feature and prediction spaces separately. Additionally, we introduce an effective Multi-Task Conditioning strategy to enhance denoising performance and facilitate learning of unlabelled tasks through cross-task information sharing. Extensive experiments on three prevalent datasets validate our approach, which outperforms previous methods by a significant margin.

Acknowledgement This research is partially supported by the Early Career Scheme of the Research Grants Council (RGC) of the Hong Kong SAR under grant No. 26202321.

References

[1] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *CVPR*, 2019. 1, 2, 6

[2] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *CVPR*, 2020. 2

[3] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017. 2

[4] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, 2018. 1, 2, 3

[5] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *TPAMI*, 44(7):3614–3633, 2021. 1, 2

[6] Wei-Hong Li, Xialei Liu, and Hakan Bilen. Learning multiple dense prediction tasks from partially annotated data. In *CVPR*, 2022. 1, 2, 6, 7

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2, 3, 4

[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 2, 3, 5

[9] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016. 2

[10] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *CVPR*, 2019. 6

[11] Yu Zhang and Qiang Yang. A survey on multi-task learning. *TKDE*, 2021.

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

[13] Menelaos Kanakis, Thomas E Huang, David Brüggemann, Fisher Yu, and Luc Van Gool. Composite learning for robust and effective dense predictions. In *WACV*, 2023.

[14] Hanxue Liang, Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Kai Zou, Yu Cheng, Cong Hao, and Zhangyang Wang. M3 vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. In *NeurIPS*, 2022.

[15] Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik Learned-Miller, and Chuang Gan. Mod-squad: Designing mixture of experts as modular multi-task learners. *arXiv preprint arXiv:2212.08066*, 2022.

[16] Lukas Hoyer, Dengxin Dai, Yuhua Chen, Adrian Koring, Suman Saha, and Luc Van Gool. Three ways to improve semantic segmentation with self-supervised depth estimation. In *CVPR*, 2021. 2

[17] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *ECCV*, 2020. 2, 3

[18] David Bruggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction. In *ICCV*, 2021. 2

[19] Siwei Yang, Hanrong Ye, and Dan Xu. Contrastive multi-task dense prediction. In *AAAI*, 2023. 2

[20] Shikun Liu, Stephen James, Andrew J Davison, and Edward Johns. Auto-lambda: Disentangling dynamic task relationships. *TMLR*, 2022.

[21] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018. 2

[22] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *NeurIPS*, 2021. 2

[23] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In *NeurIPS*, 2020.

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

[25] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *ICLR*, 2020.

[26] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, 2020. 2

[27] Hanrong Ye and Dan Xu. Taskprompter: Spatial-channel multi-task prompting for dense scene understanding. In *ICLR*, 2023. 2

[28] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *ECCV*, 2022.

[29] Yangyang Xu, Xiangtai Li, Haobo Yuan, Yibo Yang, and Lefei Zhang. Multi-task learning with multi-query transformer for dense prediction. *TCSVT*, 2023.

[30] Hanrong Ye and Dan Xu. Taskexpert: Dynamically assembling multi-task representations with memorial mixture-of-experts. In *ICCV*, 2023.

[31] Xiaogang Xu, Hengshuang Zhao, Vibhav Vineet, Ser-Nam Lim, and Antonio Torralba. Mtformer: Multi-task learning via transformer and cross-task reasoning. In *ECCV*, 2022.

[32] Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. In *ECCV*, 2022. 3

[33] Deblina Bhattacharjee, Tong Zhang, Sabine Süsstrunk, and Mathieu Salzmann. Mult: an end-to-end multitask learning transformer. In *CVPR*, 2022.

[34] Wei-Hong Li, Xialei Liu, and Hakan Bilen. Universal representations: A unified look at multiple task and domain learning. *arXiv preprint arXiv:2204.02744*, 2022.

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

[36] Yuan Gao, Haoping Bai, Zequn Jie, Jiayi Ma, Kui Jia, and Wei Liu. Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *CVPR*, 2020.

[37] Lijun Zhang, Xiao Liu, and Hui Guan. Automtl: A programming framework for automating efficient multi-task learning. In *NeurIPS*, 2021. 2

[38] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 3

[39] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *NeurIPS*, 2021. 3

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[41] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 3

[42] Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. Card: Classification and regression diffusion models. *arXiv preprint arXiv:2206.07275*, 2022. 3

[43] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *ICLR*, 2022. 3

[44] Tomer Amit, Eliya Nachmani, Tal Shaharbany, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021.

[45] Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton, and David J Fleet. A generalist framework for panoptic segmentation of images and videos. *arXiv preprint arXiv:2210.06366*, 2022.

[46] Zhangxuan Gu, Haoxing Chen, Zhuoer Xu, Jun Lan, Changhua Meng, and Weiqiang Wang. Diffusioninst: Diffusion model for instance segmentation. *arXiv preprint arXiv:2212.02773*, 2022.

[47] Hanrong Ye, Jason Kuen, Qing Liu, Zhe Lin, Brian Price, and Dan Xu. Seggen: Supercharging segmentation models with text2mask and mask2img synthesis. *arXiv preprint arXiv:2311.03355*, 2023. 3

[48] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. *arXiv preprint arXiv:2211.09788*, 2022. 3

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 5

[50] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 111:98–136, 2010. 6

[51] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 6

[52] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 6

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6