Universal Few-shot Learning of Dense Prediction Tasks with Visual Token Matching

Xinze LI

12 May, 2024

Background

Current problems

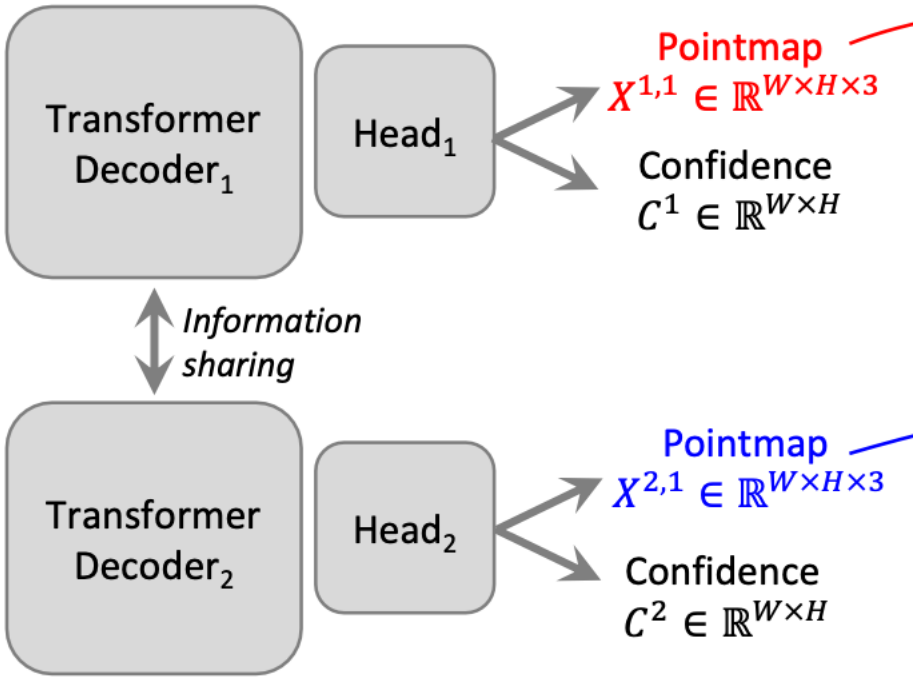
- 现有的少样本学习方法专门针对解决一组受限任务，而非密集预测任务中的所有类别。因此，它们通常在设计模型架构和训练过程中利用特定于这些任务的先验知识和假设

Advocation

- 学习器必须有一个统一的架构，并且跨任务共享大部分参数，以便它可以获得对任意未见任务的少样本学习的可推广知识。（这点跟 Dust 3r 很像）
- 学习器应该灵活地调整其预测机制，以解决未见语义的各种任务，同时足够高效，以防止过拟合。

题外话：DUSt3R: Geometric 3D Vision Made Easy

- Decoder: A generic transformer network equipped with cross attention
- self-attention (each token of a view attends to tokens of the same view
- cross-attention (each token of a view attends to all other tokens of the other view)

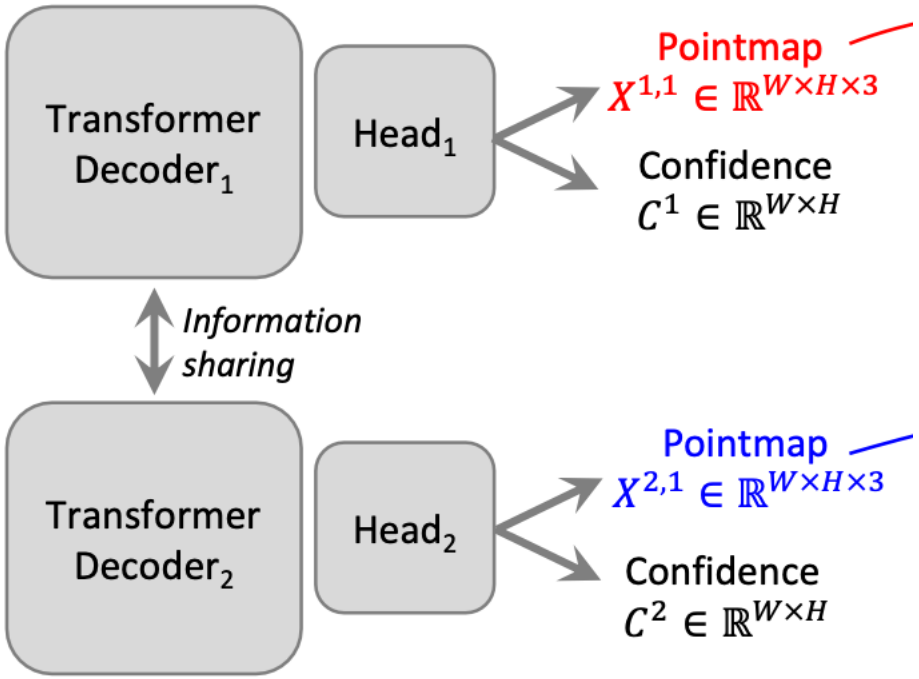


题外话：DUSt3R: Geometric 3D Vision Made Easy

- Information Sharing

$$G_i^1 = \text{DecoderBlock}_i^1\left(G_{i-1}^1, G_{i-1}^2\right),$$
$$G_i^2 = \text{DecoderBlock}_i^2\left(G_{i-1}^2, G_{i-1}^1\right),$$

for $i = 1, \ldots, B$ for a decoder with B blocks and initialized with encoder tokens $G_0^1 := F^1$ and $G_0^2 := F^2$. Here, $\text{DecoderBlock}_i^v(G^1, G^2)$ denotes the i-th block in branch $v \in \{1, 2\}$, G^1 and G^2 are the input tokens, with G^2 the tokens from the other branch. Finally, in each branch a sep-



Novelty

- 提出视觉令牌匹配（VTM），这是一个针对任意密集预测任务的通用少样本学习器。
- 在 VTM 中，analogy making for dense prediction 实现为基于 patch-level 而非参数匹配，其中模型学习图像补丁之间的相似性，以捕获标签补丁之间的相似性。（这个地方值得讨论，为什么要设计成 patch-level，为什么 work?）

Motivation

- Analogy making: 在给定一个新任务的几个示例时，人类可以快速理解如何基于示例之间的相似性来将输入和输出联系起来（即，将相似的输出分配给相似的输入），同时灵活地根据给定的上下文改变相似性的概念。

Method

统一定义 Dense Prediction Task

- 考虑任意任务 T 可以按照下面的方式表达：

$$\mathcal{T}: \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{H \times W \times C_{\mathcal{T}}}, \quad C_{\mathcal{T}} \in \mathbb{N}.$$

- H和W：输入图像的高度和宽度。
- C_T 表示输出通道数。
- 不同的密集预测任务可能涉及不同数量的输出通道和不同的属性，例如语义分割任务的多通道二进制输出和深度估计任务的单通道连续值输出。我们的目标是建立一个通用的 Few-Shot 学习器 F，对于任何这样的任务 T，可以为一个看不见的图像（查询）X^q 给出一些标记示例（支持集）S_T 的预测 $\hat{Y}^q$：

$$\hat{Y}^q = \mathcal{F}(X^q; \mathcal{S}_{\mathcal{T}}), \quad \mathcal{S}_{\mathcal{T}} = \{(X^i, Y^i)\}_{i \leq N}.$$

统一定义 Dense Prediction Task

- 为了构建这样一个通用的少样本学习器F，我们采用传统的情景训练协议，其中训练由多个情景组成，每个情景模拟一个少样本学习问题。
- 为此，我们利用一个包含多种密集预测任务的标注示例的元训练数据集D_{train}。每个训练情景模拟数据集中特定任务 T_{train}的少样本学习场景——目标是给定支持集，为查询图像生成正确的标签。通过体验多个少样本学习情景，模型预期能够学习到用于快速灵活适应新任务的一般知识。
- 在测试时，模型被要求在训练数据集（D_{train}）中未包含的任意未见任务T_{test}上执行少样本学习。

一个 challenge

$$\mathcal{T} : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^{H\times W\times C_{\mathcal{T}}}, \quad C_{\mathcal{T}} \in \mathbb{N}.$$

- 该式中 C_{T} 的不一致导致设计一个适用于所有任务的单一、统一的模型参数化变得困难。
- Solution：把 T 拆成 C_T 个单通道输出的子任务 $T_1, \ldots, T_{C_T}$ 每一个 T 都单独使用该式定义的学习器。

$$\hat{Y}^q = \mathcal{F}(X^q; \mathcal{S}_{\mathcal{T}}), \quad \mathcal{S}_{\mathcal{T}} = \{(X^i, Y^i)\}_{i\le N}.$$

一个 challenge

- 多通道信息通常是有益的，但我们观察到它在实践中的影响微乎其微，而按通道分解引入了其他有用的好处，例如增加元训练中任务的数量、灵活地推广到未见任务的任意维度，以及在任务内部和跨任务之间更有效的参数共享。

在设计统一架构前的挑战

- Task-Agnostic Architecture
  - 之前的架构：单一任务，依赖的策略如：class prototype or binary masking, as they rely on the discrete nature of categorical labels
  - 但是 Label Space (T_{test}) 既可以是离散的也可以是连续的
  - 设计要求：允许学习器获取针对少样本学习任何未见任务的通用知识，因为它使得大部分模型参数在元训练和测试中跨所有任务共享。
- Adaptation Mechanism
  - Premise:不同语义的任务可能需要不同的特征集合. Eg.深度估计需要 3D 场景理解，而边缘估计更喜欢低级图像梯度。
  - 设计要求：根据支持集 S_T 对给定任务 T 进行特征适应。还要注意不能过拟合。

Visual Token Matching (VTM)

$$\hat{Y}^q = \mathcal{F}(X^q; \mathcal{S}_\mathcal{T}), \quad \mathcal{S}_\mathcal{T} = \{(X^i, Y^i)\}_{i \leq N}.$$

- 先来说说 motivation
- 寻求一个通用的函数形式，它使用统一的框架生成任意任务的结构化标签。
- 采用了一个 patch-level 的非参数方法，其中查询标签是通过支持标签的加权组合获得的。
- 实际上，这是对 Matching Network 的一个泛化。学习新任务=优化θ_T

denote an image (or label) on patch grid of size $M = h \times w$, where $\mathbf{x}_j$ is j-th patch. Given a query image $X^q = \{\mathbf{x}_j^q\}_{j \leq M}$ and a support set $\{(X^i, Y^i)\}_{i \leq N} = \{(\mathbf{x}_k^i, \mathbf{y}_k^i)\}_{k \leq M, i \leq N}$ for a task $\mathcal{T}$, we project all patches to embedding spaces and predict the query label $Y^q = \{\mathbf{y}_j^q\}_{j \leq M}$ patch-wise by,

$$g(\mathbf{y}_j^q) = \sum_{i \leq N} \sum_{k \leq M} \sigma\left(f_\mathcal{T}(\mathbf{x}_j^q), f_\mathcal{T}(\mathbf{x}_k^i)\right) g(\mathbf{y}_k^i), \tag{3}$$

where $f_\mathcal{T}(\mathbf{x}) = f(\mathbf{x}; \theta, \theta_\mathcal{T}) \in \mathbb{R}^d$ and $g(\mathbf{y}) = g(\mathbf{y}; \phi) \in \mathbb{R}^d$ correspond to the image and label encoder, respectively, and $\sigma : \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$ denotes a similarity function defined on the image patch embeddings. Then, each predicted label embedding can be mapped to a label patch $\hat{\mathbf{y}}_j^q = h(g(\mathbf{y}_j^q))$ by introducing a label decoder $h \approx g^{-1}$.

Architecture

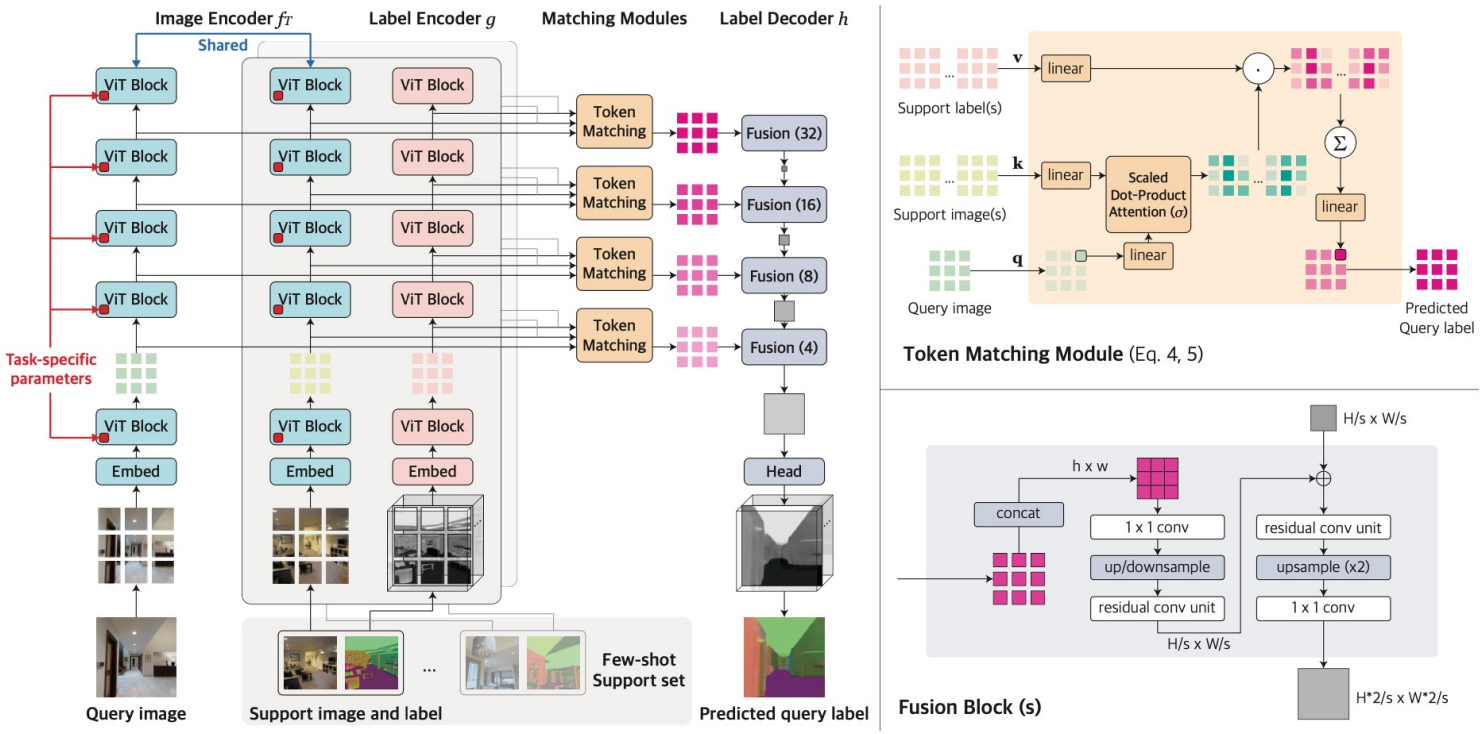


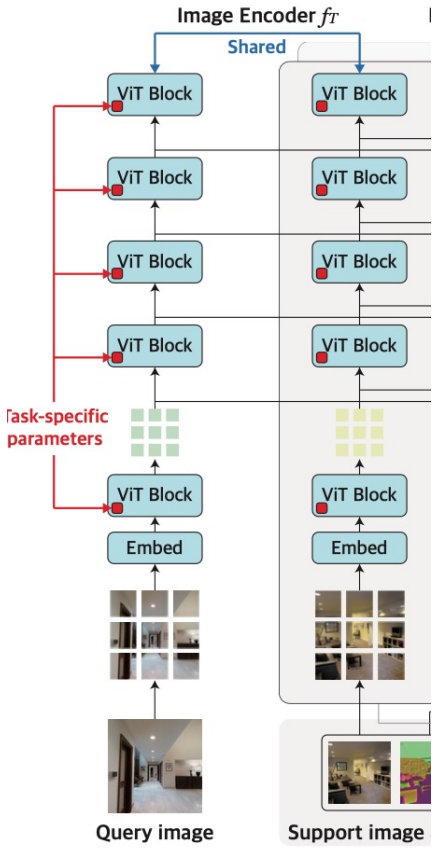
Figure 1: Overall architecture of VTM. Our model is a hierarchical encoder-decoder with four main components: the image encoder $f_{\mathcal{T}}$, label encoder g, label decoder h, and the matching module. See the text for more detailed descriptions.

流程

- 给定 Query image 和 support set，image encoder 先分别提取 patch-level 的编码 (token)
- 类似地，Label encoder 提取 support labels
- 之后，每一层都根据下式执行非参数匹配，最后生成每一个 query label 的 token

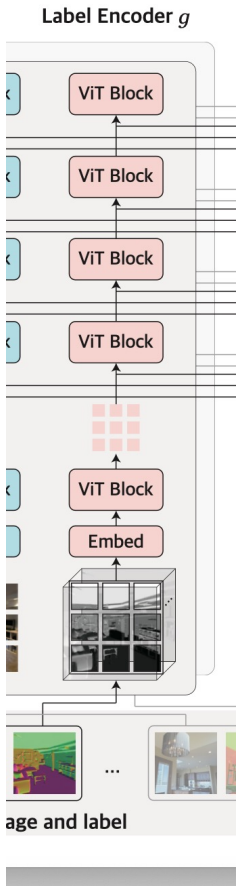
$$g(\mathbf{y}_j^q) = \sum_{i \le N} \sum_{k \le M} \sigma\left(f_{\mathcal{T}}(\mathbf{x}_j^q), f_{\mathcal{T}}(\mathbf{x}_k^i)\right) g(\mathbf{y}_k^i),$$

Image Encoder



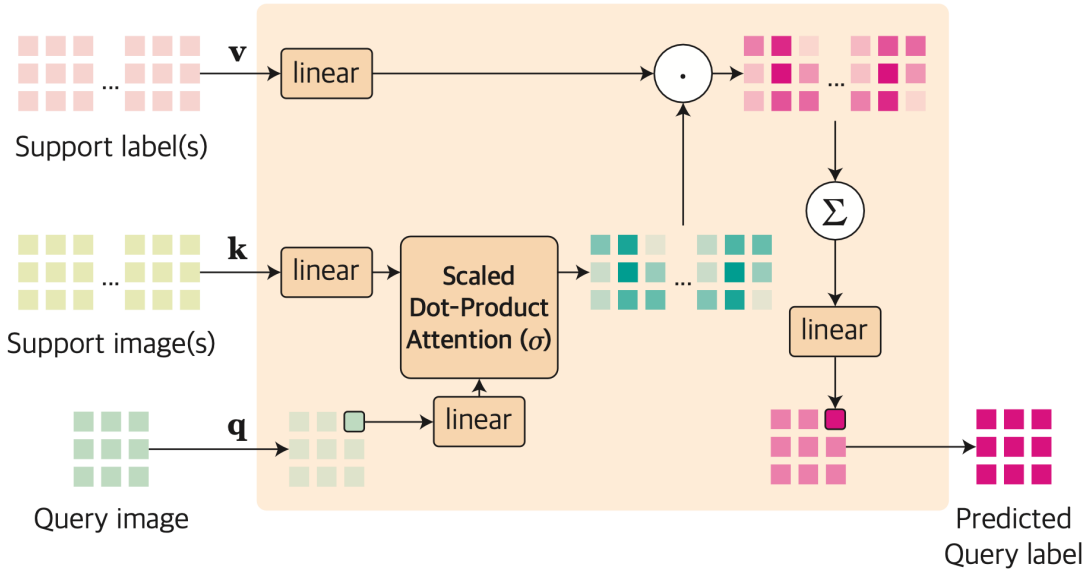
- 每一层都是 ViT
- θ_T 使用 bias tunning 来实现适应机制

Label Encoder



- 跟 Image Encoder 基本一样
- 但主要区别在于，由于将每个通道视为独立的任务，因此它一次只能处理一个通道。然后，从与图像编码器匹配的多个层次中提取标签令牌。与图像编码器相反，标签编码器的所有参数都是从头开始训练的，并且在所有任务之间共享。

Matching Module



Token Matching Module (Eq. 4, 5)

- MHA 来实现匹配，咋定义的略过了。
- MHA符合非参数匹配的直觉：因为每个query label token 都被推断为support label token v 的加权组合，基于query and support image token q, k之间的相似性。
- 相似性：scaled dot-product attention

Training and Inferencing

- 不讲了，感觉没时间了

Experiment

Quantitative comparison

- semantic segmentation (SS)
- surface normal (SN)
- Euclidean distance (ED)
- Z-buffer depth (ZD)
- texture edge (TE)
- occlusion edge (OE)
- 2D keypoints (K2)
- 3D keypoints (K3)
- reshading (RS)
- principal curvature (PC).

Quantitative comparison

Table 1: Quantitative comparison on Taskonomy dataset. Few-shot baselines are 10-shot evaluated on each fold after being trained on the tasks from the other folds, where fully-supervised baselines are trained and evaluated on tasks from each fold (DPT) or all folds (InvPT).

Supervision	Model	Tasks									
		Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
		SS	SN	ED	ZD	TE	OE	K2	K3	RS	PC
		mIoU ↑	mErr ↓	RMSE ↓	RMSE ↓	RMSE ↓	RMSE ↓	RMSE ↓	RMSE ↓	RMSE ↓	RMSE ↓
Full	DPT	**0.4449**	**6.4414**	**0.0534**	**0.0268**	**0.0188**	**0.0689**	**0.0358**	**0.0357**	**0.0860**	**0.0347**
	InvPT	0.3900	12.9249	0.0589	0.0298	0.0517	0.0788	0.0456	0.0384	0.0949	0.0370
10-Shot ($< 0.004\%$)	HSNet	0.1069	24.9120	0.2375	0.0748	0.1746	0.1643	0.1056	0.0651	0.2627	0.0610
	VAT	0.0353	25.8134	0.2718	0.0779	0.1719	0.1655	0.1450	0.0678	0.2709	0.0796
	DGPNet	0.0261	29.1668	0.4579	0.2846	0.1881	0.2130	0.1104	0.1308	0.3680	0.3574
	Ours	**0.4097**	**11.4391**	**0.0741**	**0.0316**	**0.0791**	**0.0912**	**0.0639**	**0.0519**	**0.1089**	**0.0420**

Quantitative comparison

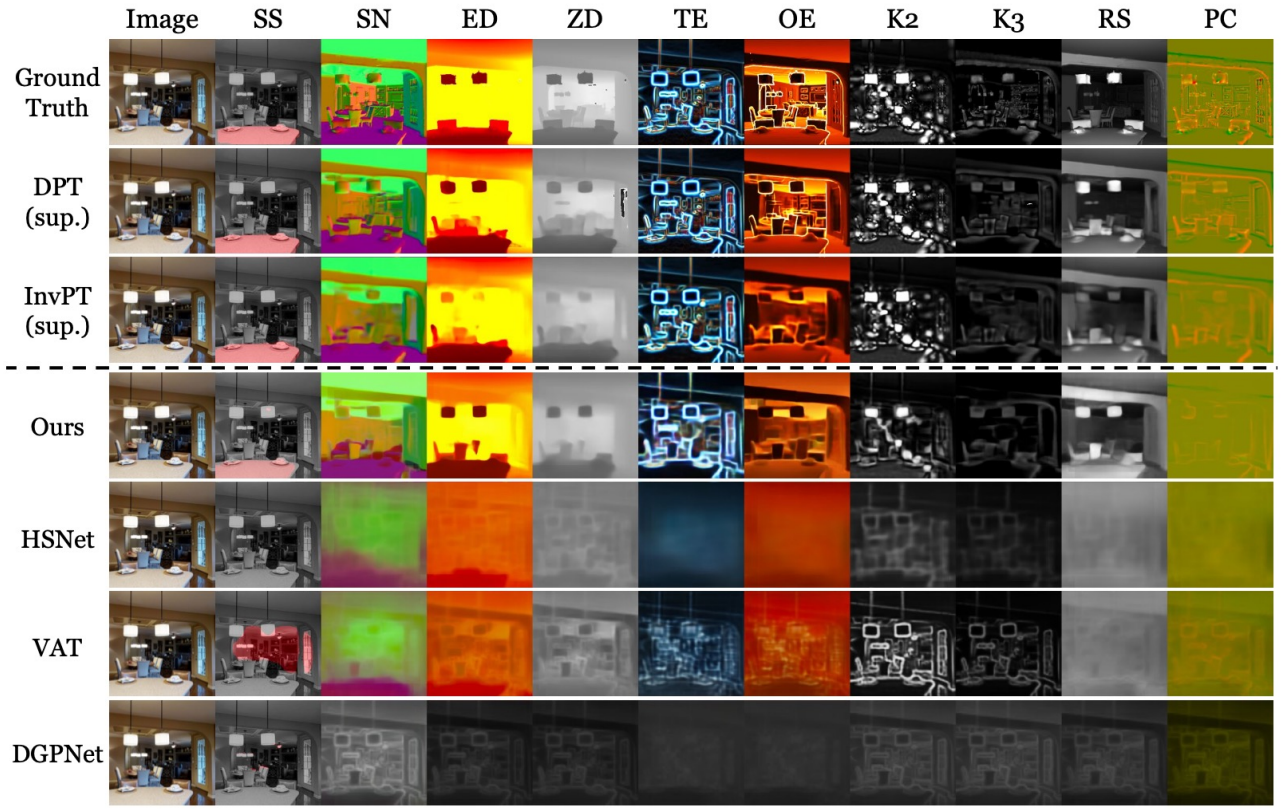


Figure 11: Additional results of qualitative comparison between Ours and the baselines.

Performance of VTM on various shots

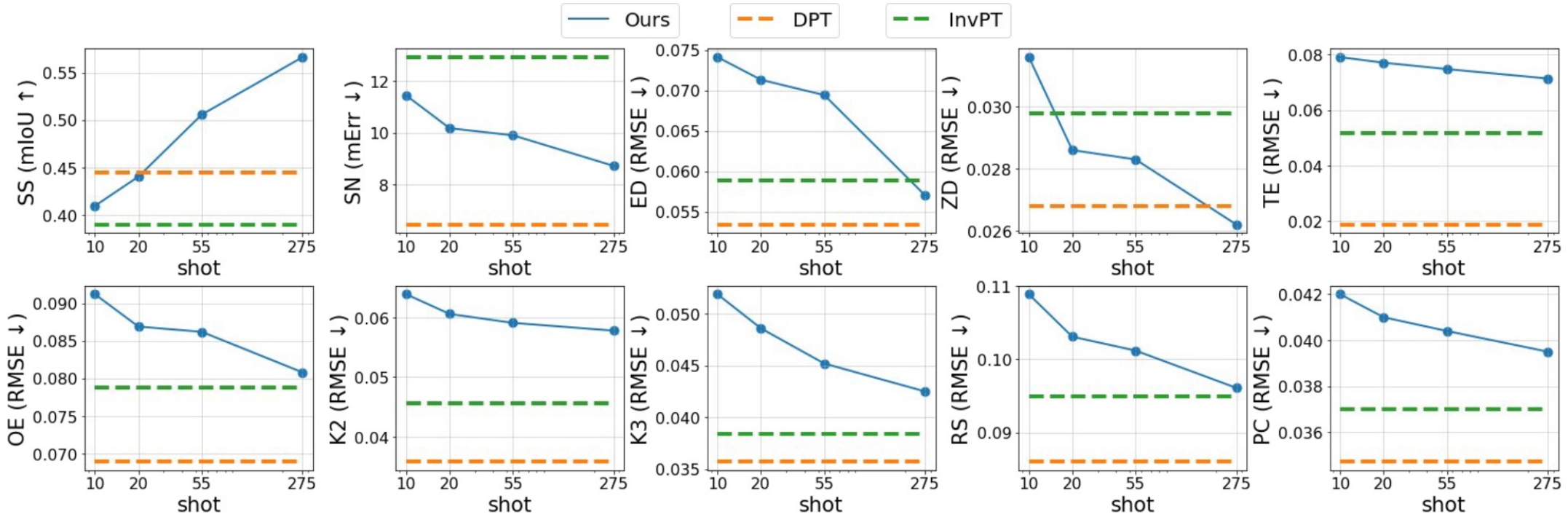


Figure 3: Performance of VTM on various shots. In general, VTM consistently improves performance as more supervision is given, and even surpasses fully supervised baselines on many tasks.

Ablation study

Table 2: Ablation study on matching and task-specific adaptation.

Model	Tasks									
	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
	SS	SN	ED	ZD	TE	OE	K2	K3	RS	PC
	mIoU ↑	mErr ↓	RMSE ↓	RMSE ↓	RMSE ↓	RMSE ↓	RMSE ↓	RMSE ↓	RMSE ↓	RMSE ↓
Ours w/o Matching	0.2681	13.0704	0.1111	0.0404	**0.0778**	0.1061	**0.0613**	0.0537	0.1559	0.0445
Ours w/o Adaptation	0.0002	23.4212	0.1515	0.0641	0.1513	0.1152	0.1110	0.0625	0.2033	0.0632
Ours	**0.4097**	**11.4391**	**0.0741**	**0.0316**	0.0791	**0.0912**	0.0639	**0.0519**	**0.1089**	**0.0420**

Evaluation