Spatially Guiding Unsupervised Semantic Segmentation Through Depth-Informed Feature Distillation and Sampling

Leon Sick
Ulm University
leon.sick@uni-ulm.de

Dominik Engel
Ulm University
dominik.engel@uni-ulm.de

Pedro Hermosilla
TU Vienna
phermosilla@cvl.tuwien.ac.at

Timo Ropinski
Ulm University
timo.ropinski@uni-ulm.de

Abstract

Traditionally, training neural networks to perform semantic segmentation required expensive human-made annotations. But more recently, advances in the field of unsupervised learning [11] have made significant progress on this issue and towards closing the gap to supervised algorithms. To achieve this, semantic knowledge is distilled by learning to correlate randomly sampled features from images across an entire dataset. In this work, we build upon these advances by incorporating information about the structure of the scene into the training process through the use of depth information. We achieve this by (1) learning depth-feature correlation *by spatially correlate the feature maps with the depth maps to induce knowledge about the structure of the scene and (2) implementing farthest-point sampling to more effectively select relevant features by utilizing 3D sampling techniques on depth information of the scene. Finally, we demonstrate the effectiveness of our technical contributions through extensive experimentation and present significant improvements in performance across multiple benchmark datasets.*

1. Introduction

Semantic segmentation plays a critical role in many of today's vision systems in a multitude of domains. These include, among others, autonomous driving, retail applications, face recognition, and many more [7, 17, 23, 27, 28]. Until recently, the main body of research in this area was focused on supervised models that require a large amount pixel-level annotations for training. Not only is sourcing this image data often an effortful process, but also annotating the large datasets required for good performance comes at a high price. Several benchmark datasets report their annotation times. For example, the MS COCO dataset [18] required more than 28K hours of human annotations for around 164K images, and annotating a single image in the Cityscapes dataset [9] took 1.5 hours on average. These costs have triggered the advent of unsupervised semantic segmentation [8, 11, 13, 25], which aims to remove the need for labeled training data in order to train segmentation models. Recently, work by Hamilton et al. [11] have accelerated the progress towards removing the need for labels to achieve good results on semantic segmentation tasks. Their model, STEGO, uses a DINO-pretrained [6] Vision Transformer (ViT) [10] to extract features that are then distilled across the entire dataset to learn semantically relevant features, using a contrastive learning approach. The to-be-distilled features are sampled randomly from feature maps calculated from the same image, k-NN matched images as well as other negative images. Seong et al. [25] build on this process by trying to identify features that are most relevant to the model by discovering hidden positives. Their work exposes an inefficiency of random sampling in STEGO as hidden positives sampling leads to significant improvements. But both approaches only operate in the pixel space and therefore fail to take into account the spatial layout of the scene. Not only do we human perceive the world in 3D, but also previous work [5, 12, 26] has shown that supervised semantic segmentation can benefit greatly from spatial information during training. Inspired by these observations, we propose to incorporate spatial information in the form of depth maps into the STEGO training process. Depth is considered a product of vision and does not provide a labeled training signal. To obtain depth information for the benchmark image datasets in our evaluations, we make use of ZoeDepth [3], an off-the-shelf zero-shot monocular depth estimator to obtain spatial information of the scene.

With our method, *DepthG*, we propose to **(1)** guide the model to learn a rough spatial layout of these scene, since

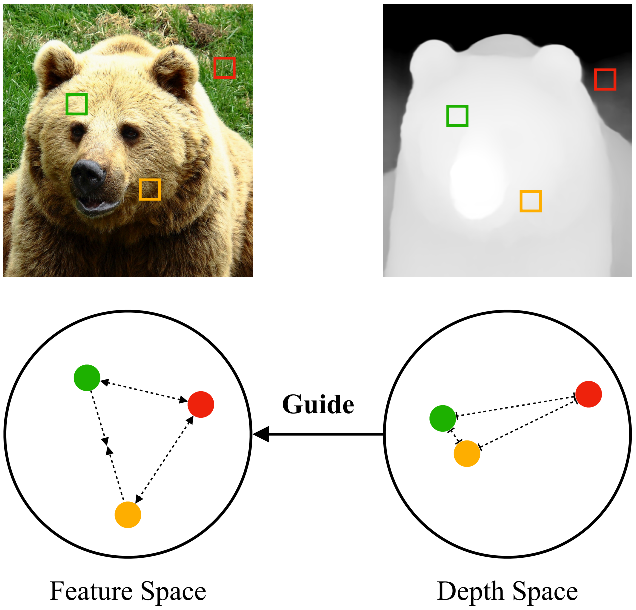

Figure 1. **Guiding the feature space for unsupervised segmentation with depth information.** Our intuition behind the proposed approach is simple: For locations in the 3D space with a low distance, we guide the model to map their features closer together. Vice versa, the features are learned to be drawn apart in feature space if their distance in the metric space is large.

we hypothesize this will aid the network in differentiating objects much better. We achieve this by extending the contrastive process to the spatial dimension: We do not limit the model to learning only Feature-Feature Correlations, but also *Depth-Feature Correlations*. Through this process, the model is guided towards pulling apart the features with high distances in the feature and also the 3D space, as well as mapping them closer together if their distance is low in feature and depth space.

With the information about the spatial layout of the scene present, we furthermore propose to **(2)** spatially inform our features sampling process by utilizing *Farthest-Point Sampling (FPS)* [21] on the depth map, which equally samples scenes in 3D. We show that this is beneficial for unsupervised segmentation, since for our evaluations on COCO-Stuff [4], we demonstrate state-of-the-art performance with *33% fewer feature samples* per image compared to STEGO.

To the best of our knowledge, we are the first to propose a mechanism to incorporate 3D knowledge of the scene into unsupervised learning for 2D images *without* encoding depth maps as part of the network input. This alleviates the risk of the model developing an input dependency, where its performance degrades at inference time since depth information is no longer available. Our approach does not rely on depth information during inference.

2. Related Work

2.1. Unsupervised Semantic Segmentation

Recent works [8, 11, 13, 25] have attempted to tackle semantic segmentation without the use of human annotations. Ji et al. [13] propose IIC, a method that aims to maximize the mutual information between different augmented versions of an image. PiCIE, published by Cho et al. [8], introduces an inductive bias made up of the invariance to photometric transformations and equivariance to geometric manipulations. DINO [6] often serves as a critical component to unsupervised segmentation algorithms, since the self-supervised pre-trained ViT can produce semantically relevant features. Recent work by Seitzer et al. [24] build upon this ability by training a model with slot attention [20] to reconstruct the feature maps produced by DINO from the different slots. The features of their object-centric model are clustered with k-means [19] where each slot is associated with a cluster. In their 2021 work, Hamilton et al. [11] have also built upon DINO features by introducing a feature distillation process with features from the same image, k-NN retrieved examples as well as random other images from the dataset. Their learned representations are finally clustered and refined with a CRF [15] for semantic segmentation. While STEGO's feature selection process is random, Seong et al. [25] introduce a more effective sampling strategy by discovering hidden positives. During training, they form task-agnostic and task-specific feature pools. For an anchor feature, they then compute the maximum similarity to any of the pool features and sample locations in the image have greater similarity than the determined value. A more detailed introduction to both latter works is provided in Section 3.1.

2.2. Depth For Semantic Segmentation

Previous research [5, 12, 26] has sought to incorporate depth for semantic segmentation in different settings. Wang et al. [26] propose to use depth for adapting segmentation models to new data domains. Their method adds depth estimation as an auxiliary task to strengthen the prediction of segmentation tasks. Furthermore, they approximate the pixel-wise adaption difficulty from source to target domain through the use of depth decoders. Work by Hoyer et al. [12] explores three further strategies of how depth can be useful for segmentation. First, they propose using a shared backbone to share learning features for segmentation and self-supervised depth estimation, similar to Wang et al. [26]. Second, they use depth maps to introduce a data augmentation that is informed by the structure of the scene. And lastly, they detail the integration of depth into an active learning loop as part of a student-teacher setup.

3. Method

In the following, we detail our proposed method for guiding unsupervised segmentation with depth information. An overview of our technique is presented in Figure 2.

3.1. Preliminary

Our approach builds upon work by Hamilton et al. [11]. In their work, each image is 5-cropped and k-NN correspondences between these images are calcualted using the DINO ViT [6]. Generally, STEGO uses a feature extractor $\mathcal{F}$ to calculate a feature map $f \in \mathbb{R}^{C \times H \times W}$ with height H, width W and feature dimension C of the input image. These features are then further encoded by a segmentation head $\mathcal{S}$ to calculate the code space $g \in \mathbb{R}^{C \times I \times J}$ with code dimension C. With the goal of forming compact clusters and amplifying the correlation of the learned features, let f and g be feature maps for a given input pair of x_i and y_i, which are then used to calculate $s := \mathcal{S}(f)$ and $q := \mathcal{S}(g)$ from the segmentation head $\mathcal{S}$. In practice, STEGO samples N^2 vectors from the feature map during training. Hamilton et al. [11] introduced the concept of constructing the feature correspondence tensor as follows:

$$\boldsymbol{F}_{hw,ij} = \frac{f_{hw} \cdot g_{ij}}{\|f_{hw}\| \|g_{ij}\|} \tag{1}$$

where $\cdot$ denotes the dot product. After the same computation for s and q, we get $\boldsymbol{S}_{hw,ij}$. Consequently, the feature correlation loss is defined as:

$$\mathcal{L}_{\text{Corr}} := -\sum_{hw,ij} (\boldsymbol{F}_{hw,ij} - b) \max(\boldsymbol{S}_{hw,ij}, 0) \tag{2}$$

where b is a bias hyperparameter. Empirical evaluations have shown, that applying spatial centering to the feature correlation loss along with zero-clamping it further improves performance. STEGO calculates these correlations for two crops from the same image and one from a different but similar image, determined by the k-NN correspondence pre-processing. Finally, negative images are sampled randomly. The final loss is a weighted sum of the different losses where each of them has their individual weight λ_i and bias b_i:

$$\mathcal{L}_{\text{STEGO}} = \lambda_{\text{self}} \mathcal{L}_{\text{self}} + \lambda_{\text{knn}} \mathcal{L}_{\text{knn}} + \lambda_{\text{random}} \mathcal{L}_{\text{random}} \tag{3}$$

After training, the resulting feature maps for a test image are clustered and refined with a conditional random field (CRF) [15].

3.2. Depth Map Generation

Since in many cases, depth information about the scene is not readily available, we make use of recent progress in the field of monocular depth estimation [1–3, 16, 22] to obtain depth maps from RGB images. Recently, methods from this field have made significant for zero-shot depth estimation i.e., predicting depth values for scenes from data domains not seen during training. This property makes them especially suitable for our method since it enables us to obtain high-quality depth predictions for a wide variety of data domains without ever re-training the depth network. It also limits the computational cost for our method. We further discuss this aspect of our method in Section 5.2. For our method, we experiment with different state-of-the-art monocular depth estimators, and use ZoeDepth [3] in our experiments. Give an cropped RGB image x_i, we use the monocular depth estimator M to predict depth $d(x_i)_{ij} \in [0, 1]$ with:

$$d(x_i) = M(x_i) \tag{4}$$

After prediction, we transform $d(x_i)$ to be in $[0, 255]$ and downsample it to match the dimensions of the feature map.

3.3. Depth-Feature Correlation Loss

With our depth-feature correlation loss, we aim to enforce spatial consistency in the feature map by transferring the distances from the spatial layout to the latent space.

In contrastive learning, the network is incentiviced to decrease the distance in feature space for similar instances, therefore learning to map their latent representations closer together. Likewise, different instances are drawn further apart in feature distance. This can be achieved through a constrative objective such as:

$$\mathcal{L}(\mathbf{z}_i, \mathbf{z}_j) = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}$$

where $\text{sim}(\mathbf{z}_i, \mathbf{z}_j)$ computes the similarity score between two feature representations $\mathbf{z}_i$ and $\mathbf{z}_j$, and τ is a temperature parameter that controls the sharpness of the probability distribution over similarities. $\mathbb{1}_{[k \neq i]}$ is an indicator function that is 1 when $k \neq i$ and 0 otherwise.

We assume the same concept to be true in 3D space: The spatial distance between two points from the same depth plateau is smaller, while the distance between a point in the foreground and one in the background is larger. Since, in both spaces, the concept of measuring difference is represented by the distance between two points, we propose to align them through our concept of *depth-feature correlation*: For large distances in the 3D space, we guide the network to produce vectors that are further apart, and vice versa. With this, we induce the model with knowledge about the spatial structure of the scene, enabling it to better differentiate between objects in the pixel and vector space. For the depth maps, just like for features, we compute a correspondence tensor.

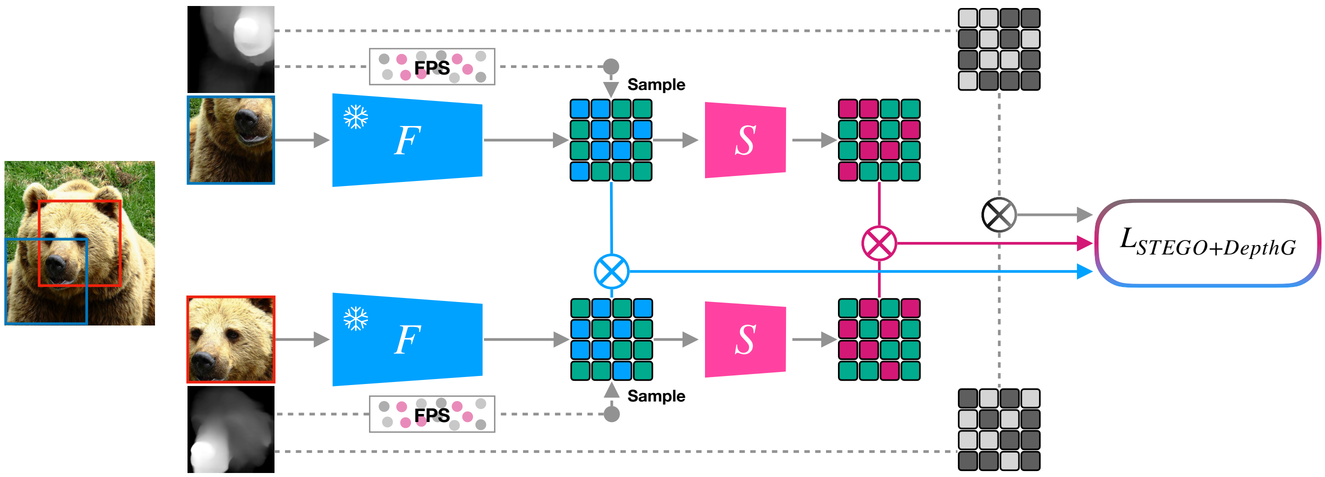

Figure 2. **Overview of the DepthG training process.** After 5-cropping the image, each crop is encoded by the DINO-pretrained ViT $\mathcal{F}$ to output a feature map. Using farthest-point-sampling (FPS), we sample the 3D space equally and convert the coordinates to select samples in the feature map. The sampled features are further transformed by the segmentation head $\mathcal{S}$. For both feature maps, the correlation tensor is computed. Following, we sample the depth map at the coordinates obtained by FPS and compute a correlation tensor in the same fashion. Finally, we compute our depth-feature correlation loss and combine it with the feature distillation loss from STEGO. We guide the model to learn depth-feature correlation for crops of the same image, while the feature distallation loss is also applied to k-NN-selected and random images.

Let $u = d(x_i)$ and $v = d(y_i)$ be the depth maps obtained for two different crops. The depth maps represent the estimated depths at each pixel of the respective image. We construct the depth correspondence tensor $\boldsymbol{D}$, defined as follows:

$$\boldsymbol{D}_{hw,ij} = u_{hw} v_{ij}, \tag{5}$$

where (h, w) and (i, j) represent the pixel positions in the depth maps u and v respectively. Together with the zero clamping, our depth-feature correlation loss is defined as:

$$\mathcal{L}_{\text{DepthG}} := -\sum_{hw,ij} (\boldsymbol{D}_{hw,ij} - b) \max(\boldsymbol{S}_{hw,ij}, 0) \tag{6}$$

where $\boldsymbol{D}_{hw,ij}$ represents the depth correlation tensor, and $\boldsymbol{S}_{hw,ij}$ represents the feature correlation tensor computed from the output features of the segmentation head $\mathcal{S}$. By also using zero-clamping, we limit erroneous learning signals that aim to draw apart instances of the same class if they have large spatial differences.

With this, we extend the STEGO loss so it can be formulated as follows:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{STEGO} + \lambda_{DepthG} \mathcal{L}_{DepthG} \tag{7}$$

with depth-feature correlation weight λ_{DepthG}. By inducing depth knowledge during training *without* encoding the depth maps as part of the model input, we alleviate the problem of the networks being at risk of depth input dependence at test time when depth input is no longer available. To the best of our knowledge, we are the first to achieve this depth distillation for unsupervised learning using only image input to the model.

3.4. Depth-Guided Feature Sampling

We also aim to make the feature sampling process informed by the spatial laypout of the scene. To perform sampling in the depth space, we transform the downsampled depth map $d(x_i)$ into a point cloud with points $\{p_1, p_2, ..., p_n\}$. On this point cloud, we apply farthest point sampling (FPS), in an iterative fashion by always selecting the next point p_{ij} as the point with the maximum distance in 3D space with respect to the rest of points $\{p_{i_1}, p_{i_2}, ..., p_{i_{j-1}}\}$. After having sampled N^2 points, we end up with a set of samples $\{p_{i_1}, p_{i_2}, ..., p_{i_{N^2}}\}$ which are consequently converted two 2D sampling indices for the feature maps f and g. In contrast to the data-agnostic random sampling applied in STEGO, our feature selection process takes into account the geometry of the input scene and more equally covers the spatial structure. This more equal sampling of depth space further increases the effectiveness of our depth feature correlation loss due to the increase diversity in selected 3D locations.

3.5. Guidance Scheduling

While our depth-feature correlation loss is effective at enriching the model's learning process with spatial information of the scene, we aim to alleviate the potential of it interfering the learning of feature correlations during model training. We hypothesize that our model most greatly benefits from depth information towards the beginning of training when its only knowledge is encoded in the features maps output by the frozen ViT backbone. To give it a head start, we increase the weight of our depth-feature correlation loss at the start and gradually decrease its influence during training. Vice versa, the knowledge distillation process in the feature space be will emphasised more strongly as the model training progresses. In this way, the network builds upon the already learned rough structure of the scene achieved through our depth guidance process. We find an exponential decay of the weight for our loss component work particularly well. Therefore, we update the weight λ_{Depth} and bias b_{Depth} every m steps according to:

$$\lambda_{\text{Depth}}(t) = \begin{cases} \lambda_{\text{Depth}}(t-1)^{\lfloor \frac{t}{m} \rfloor}, & \text{if } t > 0 \\ \lambda_{\text{Depth}}^{\text{init}} & \text{if } t = 0 \end{cases} \tag{8}$$

and

$$b_{\text{Depth}}(t) = \begin{cases} b_{\text{Depth}}(t-1)^{\lfloor \frac{t}{m} \rfloor}, & \text{if } t > 0 \\ b_{\text{Depth}}^{\text{init}} & \text{if } t = 0 \end{cases} \tag{9}$$

In practice, λ_{Depth} and b_{Depth} are never decayed to 0.

4. Experiments

4.1. Evaluation Settings

To evaluate our method, we largely follow the protocols from STEGO. [11]

Datasets and Model Sizes. We conduct experiments on the COCO-Stuff [4], Cityscapes [9], and Potsdam-3 datasets. The COCO-Stuff contains a wide variety of scenes and its class distribution can be split into 101 classes (fine) and 27 classes (coarse). In our evaluation, we follow [11, 13, 25] an provide results on the coarse class split, COCO-Stuff 27. In contrast, Cityscapes contains traffic scenes from 50 cities from a driver-like viewpoint. Lastly, the Potsdam-3 dataset is composed of aerial, top-down images from the city of Potsdam. We use the DINO [6] backbones ViT-Small (ViT-S) and ViT-Base (ViT-B) with a patch size of $8x8$, which were pre-trained in a self-supervised manner.

Evaluation Protocols. Similar to [11, 25], we evaluate our models in the unsupervised, clustering-based setting as well as the linear probe setting. Since the output of our model is a pixel-level map of features and not class labels, these features are clustered. Following, the pseudo-labeled clusters are aligned with the ground truth labels through Hungarian matching across the entire validation dataset. To perform linear probing, an additional linear layer is added on top of the model and trained with cross-entropy loss to learn classifying the produced features.

4.2. COCO-Stuff

We present our evaluation on COCO-Stuff27 in Table 1. For the ViT-S/8, our experiments show that our method is able to improve upon STEGO in most metrics, with improved unsupervised accuracy by **+8.0%** and unsupervised mIoU increased by **+1.1%**. When comparing our approach to Hidden Positives, a method with much more computational overhead, for the ViT-S/8, we show competitive performance for unsupervised accuracy and outperform their approach by **+1.0%** on unsupervised mIoU. When using the DINO ViT-B/8 encoder, our approach again outperforms STEGO as well as all other presented methods on unsupervised metrics. Most notably, we are able to increase the unsupervised mIoU by **+0.8%**. In their study on STEGO, Koenig et al. [14] observe that frozen DINO with the frozen STEGO layers on top already shows good performance for linear probing, even outperforming trained STEGO on linear mIoU.

Setting		Unsupervised		Linear	
Method	Model	Acc.	mIoU	Acc.	mIoU
IIC [13]	R18+FPN	21.8	6.7	44.5	8.4
PiCIE [8]	R18+FPN	48.1	13.8	54.2	13.9
PiCIE+H [8]	R18+FPN	50.0	14.4	54.8	14.8
STEGO [11]	ViT-S/8	48.3	24.5	74.4	38.3
STEGO + HP [25]	ViT-S/8	**57.2**	24.6	**75.6**	**42.7**
STEGO + *Ours*	ViT-S/8	56.3	**25.6**	73.7	38.9
DINO [6, 14]	ViT-B/8	42.2	13.0	75.8	44.4
DINOSAUR [24]	ViT-B/8	44.9*	24.0*	-	-
STEGO [11]	ViT-B/8	56.9	28.2	**76.1**	41.0
STEGO + *Ours*	ViT-B/8	**58.6**	**29.0**	75.5	**41.6**

Table 1. **Evaluation on COCO-Stuff-27.** We report results on COCO-Stuff with 27 high-level classes. Overall, our method outperforms STEGO and HP on unsupervised segmentation with the ViT-B/8, while showing competitive performance for the ViT-S/8. *Results from the paper obtained without post-processing optimization.

4.3. Cityscapes

Method	Model	U. Acc	U. mIoU
IIC [13]	R18+FPN	47.9	6.4
PiCIE [8]	R18+FPN	65.6	12.3
STEGO	ViT-B/8	73.2	21.0
STEGO + HP	ViT-B/8	79.5	18.4
STEGO + *Ours*	ViT-B/8	**81.6**	**23.1**

Table 2. **Results on Cityscapes.** We report unsupervised accuracy and mIoU on Cityscapes. Our method outperforms both STEGO variants by substantial margins. Notably, our method is the first to improve upon unsupervised mIoU.

We further evaluate our approach in the Cityscapes dataset [9], made up of various scenes from 50 different cities, annotated with 30 classes. As can be seen in Table 2, our method significantly outperforms STEGO as well as Hidden Positives on both metrics. For unsupervised mIoU,

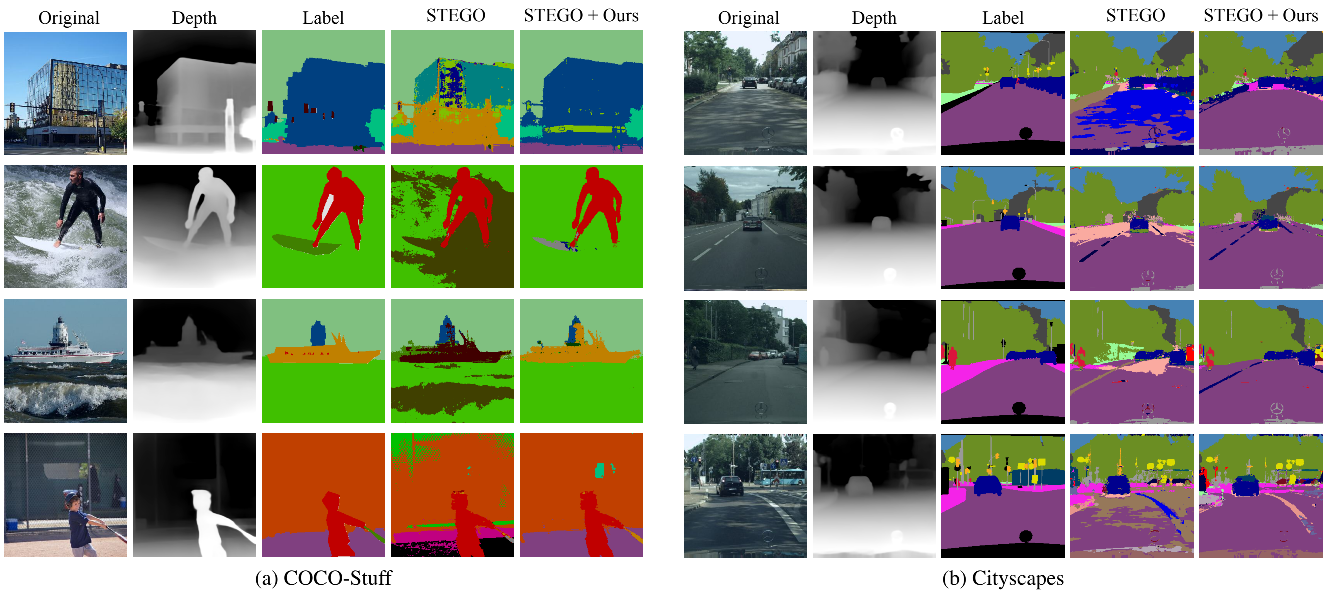

Figure 3. **Qualitative results.** We show qualitative differences for plain STEGO compared to STEGO with our depth guidance, using ViT-S models for COCO and ViT-B for Cityscapes. Where STEGO struggles to differentiate difference instances, our model is able to correct this and successfully separate them for segmentation. In the case of the the building in 3a, our method alleviates visual irritations from the pixel space and significantly correctly the segmentation of the building. For 3b Cityscapes, our model is able to better handle visual inconsistencies from shadows.

while Hidden Positives decreased performance compared to STEGO, we observe our approach to achieve a **+2.1%** increase. Similarity, we report state-of-the-art performance in accuracy, building upon Hidden Positives' already impressive improvements upon STEGO and outperforming it by **+2.1%**.

4.4. Potsdam

Method	Model	Unsupervised Accuracy
IIC [13]	R18+FPN	65.1
STEGO	ViT-B/8	77.0
STEGO + HP	ViT-B/8	**82.4**
STEGO + *Ours*	ViT-B/8	80.4

Table 3. **Results on Potsdam.** We report unsupervised accuracy on the Potsdam dataset. Our method is able to improve upon STEGO, but falls short of catching HP. We hypothesize that with a zero-shot depth estimator more suitable for aerial images, the results for our method could further improve.

Lastly, we evaluate our model on the Potsdam-3 dataset, containing aerial images of the German city of Potsdam. Contrary to the other benchmarks, which contain images in a first-person perspective, Potsdam-3 contains only birds-eye-view images, a perspective that is considered OOD for the monocular depth estimator. Despite this inherent limitation of our approach for aerial data, Table 3 we are able to demonstrate a relatively commendable performance by improving STEGO's performance but coming short of Hidden Positives.

4.5. Qualitative Results

We present qualitative results of our method in Figure 3 and compare with segmentation maps from STEGO. On multiple occasions, our depth guidance reduces erroneous predictions from the model caused by visual irritations in the pixel space. In the example of the boy with the baseball bat in Figure 3a, false classifications from STEGO are caused by shadows on the ground. Our model is able to correct this. Furthermore, it goes beyond the noisy label and also correctly classifies the glimpse of a plant that can be seen through a hole in the background. This is an indication that our model does not overfit to the depth map, since this visual cue is only observable from the pixel space.

5. Ablations

5.1. Individual Influence

We investigate the effect of our technical contributions on training our model with a ViT-S/8 backbone on COCO-Stuff 27. Our observations in Table 4 show that our depth-feature correlation loss itself already improves the performance of STEGO. This improvement is further increased through the use of FPS, which enables us to sample the

Method	U. Accuracy	U. mIoU
STEGO	24.5	48.3
+ *Depth-Feature Correlation*	24.7	51.2
+ *FPS* ($N = 9$)	**25.6**	**56.3**

Table 4. **Effect of our contributions.**

depth space more equally and therefore encourages more diversity in the depth correlation tensor $\boldsymbol{D}_{hw,ij}$. Intuitively, this sampling diversity significantly amplifies our depth-feature correlation for aligning the feature space with the depth space.

5.2. Computational Cost

Our method only leads to an insignificant increase in runtime versus the baseline STEGO model, since we solely guide the loss as well as the feature sampling and do not introduce additional layers. In contrast, the competitive method Hidden Positives [25] relies on a computationally more expensive process to select features and introduces an additional segmentation head to fill their task-specific feature pool. To keep the computational overhead of our method low, we make use of a pre-trained monocular depth estimation network with impressive zero-shot capabilities. While a task specific training of this method would increase the computational cost of our method, we consider this not a necessity, since the model is zero-shot capabilities generalize well to different scenes and domains. Therefore, in our experiments consisting of a diverse array of scenes, we do not re-train or finetune the depth estimator, and consider the additional computational cost for generating the depth maps to be negligible.

6. Limitations

While we have demonstrated our method effectiveness for many real-world cases, our method's applicability is limited in settings unsuitable for depth estimation, such as slices of CT scans and other medical data domains. Furthermore, the experiments on Potsdam-3 have shown, our method can improve unsupervised semantic segmentation despite suboptimal viewing perspectives for the monocular depth estimator, but we could not demonstrate state-of-the-art performance. We assume this represents a rare case where, for an increase in performance to be observed, the depth estimator would need to be retrained on domain-specific data. We also present failure cases of our model in Figure 4.

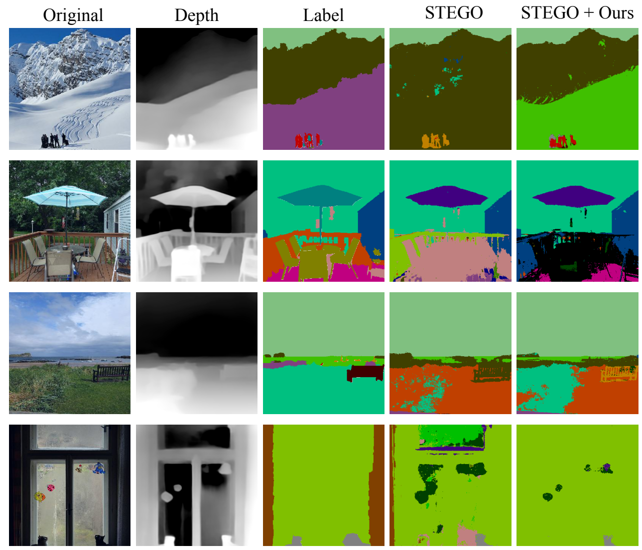

Figure 4. **Failure cases.** We show cases where our model fails to correctly segment and classify the scene. The top row is a prime example where the difference in depth is correctly distilled, though the model fails to correctly classify the snow region.

7. Conclusion & Future Work

In this work, we have presented a novel method to induce spatial knowledge of the scene into our model for unsupervised semantic segmentation. We have proposed the extension to correlate the feature space with the depth space and use the 3D information to more equally sample features in a spatially informed way. Furthermore, we have demonstrated that these contributions produce state-of-the-art performance on many real-world datasets and thus further the progress in unsupervised segmentation. The applicability of our approach for other tasks is further to be explored since we hypothesize it can be useful beyond unsupervised segmentation as part of any constrastive process. We consider this to be a promising direction for future work. Furthermore, it remains to be investigated which information could be useful to transfer our approach to medical data.

References

[1] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 3

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Localbins: Improving depth estimation by learning local distributions. In *European Conference on Computer Vision*, pages 480–496. Springer, 2022. 3

[3] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 1, 3

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 2, 5

[5] Adriano Cardace, Luca De Luigi, Pierluigi Zama Ramirez, Samuele Salti, and Luigi Di Stefano. Plugging self-supervised monocular depth into unsupervised domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1129–1139, 2022. 1, 2

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 1, 2, 3, 5

[7] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022. 1

[8] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16794–16804, 2021. 1, 2, 5

[9] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 1, 5

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 1

[11] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *International Conference on Learning Representations*, 2021. 1, 2, 3, 5

[12] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11130–11140, 2021. 1, 2

[13] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9865–9874, 2019. 1, 2, 5, 6

[14] Alexander Koenig, Maximilian Schambach, and Johannes Otterbach. Uncovering the inner workings of stego for safe unsupervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3788–3797, 2023. 5

[15] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011. 2, 3

[16] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 3

[17] Chen Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Representations*, 2015. 1

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[19] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 2

[20] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020. 2

[21] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[22] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 3

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 1

[24] M Seitzer, M Horn, A Zadaianchuk, D Zietlow, T Xiao, C Simon-Gabriel, T He, Z Zhang, B Schölkopf, Thomas Brox, et al. Bridging the gap to real-world object-centric learning. In *International Conference on Learning Representations (ICLR)*, 2023. 2, 5

[25] Hyun Seok Seong, WonJun Moon, SuBeen Lee, and Jae-Pil Heo. Leveraging hidden positives for unsupervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19540–19549, 2023. 1, 2, 5, 7

[26] Qin Wang, Dengxin Dai, Lukas Hoyer, Luc Van Gool, and Olga Fink. Domain adaptive semantic segmentation with self-supervised depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8515–8525, 2021. 1, 2

[27] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 1

[28] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmen-

tation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 1