Hierarchical Lovász Embeddings for Proposal-free Panoptic Segmentation

Tommi Kerolla1* Jie Li2 Atsushi Kanehira1 Yasunori Kudo1 Alexis Vallet1 Adrien Gaidon2
1Preferred Networks, Inc. 2Toyota Research Institute (TRI)

Abstract

Panoptic segmentation brings together two separate tasks: instance and semantic segmentation. Although they are related, unifying them faces an apparent paradox: how to learn simultaneously instance-specific and category-specific (i.e. instance-agnostic) representations jointly. Hence, state-of-the-art panoptic segmentation methods use complex models with a distinct stream for each task. In contrast, we propose Hierarchical Lovász Embeddings, per pixel feature vectors that simultaneously encode instance- and category-level discriminative information. We use a hierarchical Lovász hinge loss to learn a low-dimensional embedding space structured into a unified semantic and instance hierarchy without requiring separate network branches or object proposals. Besides modeling instances precisely in a proposal-free manner, our Hierarchical Lovász Embeddings generalize to categories by using a simple Nearest-Class-Mean classifier, including for non-instance ”stuff” classes where instance segmentation methods are not applicable. Our simple model achieves state-of-the-art results compared to existing proposal-free panoptic segmentation methods on Cityscapes, COCO, and Mapillary Vistas. Furthermore, our model demonstrates temporal stability between video frames.

1. Introduction

Holistic scene understanding is an important task in computer vision, where a model is trained to explain each pixel in an image, whether that pixel describes stuff – uncountable regions of similar texture such as grass, road or sky – or thing – a countable object with individually identifying characteristics, such as people or cars. While holistic scene understanding received some early attention [49, 55, 48], modern deep learning-based methods have mainly tackled the tasks of modeling stuff and things independently under the task names semantic segmentation and instance segmentation. Recently, Kirillov et al. proposed the panoptic quality (PQ) metric for unifying these two parallel tracks into the holistic task of panoptic segmentation [24]. Panoptic segmentation is a key step for visual understanding, with applications in fields such as autonomous driving or robotics, where it is crucial to know both the locations of dynamically trackable things, as well as static stuff classes. For example, an autonomous car needs to be able to both avoid other cars with high precision, as well as understand the location of the road and sidewalk to stay on a desired path.

A strong but complex baseline for panoptic segmentation is to run independent methods for semantic segmentation and instance segmentation, and then fusing the results. To improve upon this, previous works combine both tasks in a joint model [52, 23, 27]. Early methods focus more on a joint model and the majority of them leverage two-stage instance detection models [43, 52, 23, 29, 13]; some recent works propose bottom-up [15, 9, 53], yet instance and semantic segmentation are still treated separately. Performing panoptic segmentation as a single task without duplicated information across sub-tasks remains an interesting question.

Intuitively, instances are contained in semantics, where semantic representations have higher variance in the embedding space to describe a general category, whereas instances have smaller variance to capture object-specific characteristics. This constitutes a natural hierarchical relationship between instances and semantics (cf. Figure 4).

In this work, we propose to model panoptic segmentation as a unified task via a novel formulation of the problem: learning hierarchical pixel embeddings. Creating a unified embedding for the task opens up the potential of leverag-
wise segmentation of semantics, treating each object class as well studied [16, 14, 45]. However, previous work has not taken advantage of the semantic-instance visual hierarchy for end-to-end unified scene parsing. In this paper, we leverage advances in structural representation learning and encode “instance” and “category” features in a hierarchical embedding space. By doing so, we reduce the redundant information in the output space and optimize the information efficiency of network parameters for panoptic segmentation.

Our main contribution is a novel representation learning approach for panoptic segmentation that treats it as a unified task. We propose a simple architecture and loss to learn pixel-wise embeddings to represent instances, object categories and stuff classes, thus enabling unified embedding-based single-shot panoptic segmentation. In particular, we leverage the Lovász hinge loss to learn a structured Hierarchical Lovász Embedding Space where categories can be represented with categories jointly. An overview of our method is shown in Figure 2.

Compared to conventional panoptic segmentation models, our model displays temporal stability between video frames, creating temporal smoothness in predictions that can be directly used in downstream applications such as object tracking and prediction in autonomous driving or mobile robotic systems, where data association is a key component (cf. Figure 8). Experiments on the Cityscapes [10], COCO [30] and Vistas [38] datasets show that our method establishes state-of-the-art results for proposal-free methods, and also yields competitive results compared with two-stage models.

Figure 2. Overview of our method. We train a single-shot fully convolutional network to predict for each pixel $i$ a hierarchical embedding $e_i$ as well as an instance seed $s_i$ and variance $\sigma_i$. The seed map represents probable instance locations, and the variance defines the margins of the hierarchical embedding space. These are used for panoptic decoding of the embedding space.

2. Related Work

Deep learning-based dense prediction tasks have typically focused on either uncountable background or countable foreground objects. Semantic segmentation concerns the pixel-wise segmentation of semantics, treating each object class as uncountable. Instance segmentation, on the other hand, focuses explicitly on countable foreground classes, such as persons or cars. For the past few years, these tasks have evolved separately, with little interaction, leading to issues such as trouble with contextual clues in instance segmentation, or the confusion caused by the large variance of person classes in semantic segmentation. Recently, panoptic segmentation was proposed as a new task to bridge the gap between these methods and allow tackling them in a unified manner [24].

Embedding-based methods have recently become popular in the computer vision community for improving object detection [25, 57] and keypoint estimation [42]. In this section, we briefly review representative methods for each task.

Semantic segmentation. Following the seminal work of Long et al. [33], semantic segmentation is typically treated as a pixel-wise classification task (although exceptions exist [21]), where a fully convolutional network with an encoder-decoder architecture is trained to output a high soft-max score for the ground-truth class. There have been several improvements since then. Namely, SegNet [3] introduces unpooling layers for more accurate upsampling in the decoder. Conversely, Deeplab [7] proposed to use a network with dilated convolution instead of a decoder, and leverages pooling to capture global information. PSPNet [56] improves global context by leveraging pyramid pooling and dilated convolution. DeeplabV3+ [8] combines the advantages of pooling and encoder-decoder architectures to better capture contextual information and sharper object boundaries.

Instance segmentation. Typical high-performing instance segmentation methods are variants of the Mask R-CNN [18] framework [32, 6]. These methods work in two stages, where the first stage computes regions of interest via a region proposal network, conducts non-maximum suppression on the proposed bounding boxes, and then runs a second stage via a head network on each proposal. While yielding high accuracy, these methods are typically too slow for real-time inference. Recently, there has been work addressing the creation of accurate single-shot (proposal-free) instance segmentation methods [2, 39, 15, 50]. In particular, Neven et al. [39] showed how to accurately predict a spatial embedding space...
for instance segmentation. Unlike previous methods, their network operates in a single stage and is able to produce an instance segmentation in a single shot, while having accuracy comparable to the more expensive Mask R-CNN. Their work inspires us to explore the possibility to learn a hierarchical embedding space for panoptic segmentation.

**Panoptic segmentation.** The current dominant methods for panoptic segmentation are two-stage frameworks. Kirillov et al. [24] run PSPNet and Mask R-CNN independently to obtain semantics and instance predictions. They subsequently combine these using heuristics. Subsequent work was done to combine the independent networks into one, by adding a semantic segmentation branch to Mask R-CNN [23, 43, 13, 29, 26, 27], but manual heuristics still remained. Other works aim to further remove manual merging heuristics of the semantics and instance predictions. E.g. UPSNet [52] proposes a panoptic head network for merging the predictions, and Liu et al. [31] leverages a spatial ranking module to conduct the merging between the two branches. Yang et al. [54] propose to resolve overlaps via instance relation reasoning. Recently, there has been some work adding a mask head on-top of a transformer to avoid heuristics [5].

Single-shot approaches, while less explored [33, 15, 9, 20], are an important complementary direction for panoptic segmentation. These methods demonstrate their potential in both network accuracy [9] and inference efficiency [20]. These existing methods feature separated feature streams for semantic segmentation- and instance-related representation, while in our approach, a unified embedding is used to model both semantic (category) and instance information. That is, our method has in fact the same downstream features showing that this pixel is both a car (instance-agnostic), and also this car (instance-specific), creating a natural and inherent feature representation for panoptic segmentation.

**Embeddings for computer vision.** Associative embeddings [41, 42] and variants have been popular for various vision tasks. The embedding space is learned by leveraging push and pull forces between embeddings in the image, depending on ground-truth annotation. The initial paper on associative embeddings [42] used 1-dimensional embeddings. The follow-up work [41] showed that increasing the embedding to be 8-dimensional improved convergence of the network. Embeddings have also been used for lane detection [40], instance segmentation [11, 39], and for instance and semantic segmentation of point clouds [51]. Further, embeddings have been used with success for improving object detection methods, using embedding space as a way to remove human-designed priors [25, 57]. Our proposed method is a variation of a deep nearest class mean (NCM) classifier [17], which has shown some promising results on image classification for replacing typical networks with softmax outputs directly after the last convolutional layer. To the best of our knowledge, our work is the first to leverage a deep NCM method for semantics in panoptic segmentation.

3. Panoptic Segmentation with Hierarchical Embeddings

In this section, we provide detailed discussion on how we learn the proposed embedding space and use the predicted embeddings for panoptic segmentation, based on the framework depicted in Figure 2. We start with problem formulation, and a discussion over the limitation of a current widely used loss in embedding learning for scene parsing. Then we propose the usage of a better alternative to the task with our novel loss for hierarchical embedding space learning. Finally, we describe panoptic decoding on the learned embedding space for creating a panoptic segmentation.

3.1. Problem Formulation

We formulate the task of panoptic segmentation as an embedding space learning problem, where we want to associate each pixel $x_i$ with a latent embedding $e_i$ such that we can decode both the object instance and semantic class correctly from the same single embedding (cf. Figure 4). Our goal is to learn a function $f$ that maps a set of pixels $X$, $|X| = N$, into a set of embeddings $E$ such that the embeddings can be partitioned into sets $S$ and $I$, where $S = \{S_1, \ldots, S_{|C|}\}$ are the embedding spaces defined by the set of semantic classes $C$, and $I = \{I_1, \ldots\}$ is the embedding subspace defined by the sets of instances such that we have the hierarchy $I_l \subseteq S_k \subseteq E$ for any instance $l$ with semantic class $k$. Further, the semantic classes $C$ are assumed to be divided into thing classes $C_{\text{thing}}$ and stuff classes $C_{\text{stuff}}$. Note that we consider stuff classes to consist of a single instance.

3.2. Baseline – Associative Embeddings

A popular approach for learning such an embedding is the associative embedding (AE) loss proposed by Newell et al [42] (also known as discriminative loss [11]). The loss uses pull and push terms to attract or repel an embedding $e_i$ from or to others $e_j$, depending on the ground-truth pixel label of each embedding, given $L$ instances:

$$L_{\text{pull}} = \frac{1}{L} \sum_{l=1}^{L} \frac{1}{|I_l|} \sum_{j \in I_l} [\|\hat{e}_l - e_j\| - \delta_{\text{pull}}]^2, \quad \text{(1)}$$

$$L_{\text{push}} = \frac{1}{L(L-1)} \sum_{l \neq k} [\delta_{\text{push}} - [\|\hat{e}_l - \hat{e}_k\|]_+]^2, \quad \text{(2)}$$

where $\hat{e}_l = \frac{1}{|I_l|} \sum_{j \in I_l} e_j$ is the instance mean embedding, and $\delta_{\text{pull}}, \delta_{\text{push}}$ are problem-specific hinge hyperparameters and $[\cdot]_+$ is the ReLU function.

In theory, a set of two AE losses can be used to learn a hierarchical embedding space by enforcing the hinge hyperparameters to be larger for semantic ground-truth than for
instance ground-truth. While this idea works for toy problems, the disadvantage is that we need to manually define the hinge hyperparameters which employ additional uncertainty and sensitivity in real-world datasets. We have found this difficult to do in practice for real-world datasets such as Cityscapes [10], as depicted in our ablation analysis in Section 4.4.

3.3. Lovász Hinge Loss

To decrease the requirement for additional hinge hyperparameters, we look into the Lovász hinge loss [4], which acts as a differentiable surrogate for the intersection over union (IoU), a common measure of mask overlap. In panoptic quality (PQ) [24] evaluation, IoU is a common base metric across thing and stuff classes.

First, we briefly review the Lovász hinge loss. Given a vector of binary ground-truth labels \( t \in \{0, 1\}^N \) and predicted labels \( y \in \{0, 1\}^N \), the IoU is defined as
\[
\text{IoU}(y, t) = \frac{\left| \{ i : y_i = 1 \cap t_i = 1 \} \right|}{\left| \{ i : y_i = 1 \cup t_i = 1 \} \right|},
\]
which is a score between 0 and 1, where higher is better.

As the IoU is a discrete set function, it cannot be optimized via gradient descent, however the Lovász hinge provides a way of creating a continuous and differentiable surrogate function in terms of the prediction errors \( \xi \) for each pixel [4]. Given a continuous prediction vector \( p \in [0, 1]^N \), the binary version of the Lovász hinge loss is defined as
\[
\text{LovászBinary}(p, t) = \sum_i \xi_i \Delta \text{IoU},
\]  
with prediction error \( \xi \) and IoU difference \( \Delta \text{IoU} \) defined as
\[
\xi_i = \begin{cases} 
1 - p_i & t_i = 1 \\
1 & t_i = 0
\end{cases},
\]
\[
\Delta \text{IoU}_i = \text{IoU}(\pi_1, \ldots, \pi_{i-1}) - \text{IoU}(\pi_1, \ldots, \pi_i),
\]
\( \pi \) = indices that sort \( \xi \) in descending order.

Here, with slight abuse of notation, \( \text{IoU}(\pi_1, \ldots, \pi_i) \) denotes calculating IoU given the \( i \) indices with the largest prediction error \( \xi_i \). For multi-class prediction, we apply the binary loss in an all-vs-one manner:
\[
\text{LovászSoftmax}(p, t) = \frac{1}{N} \sum_{i \in C} \text{LovászBinary}(p^t, t^i),
\]
where \( t^i = 1 \) if the ground-truth class is \( c \), and 0 otherwise.

3.4. Hierarchical Lovász Embeddings

We propose to leverage the Lovász hinge loss to learn a shared embedding space for both semantics and instances. Given an instance \( I_i \), with mean \( \mu_i \) and variance \( \sigma_i \), the score of an embedding \( e_i \in \mathbb{S}^{d-1} \) lying on the unit hypersphere belonging to instance \( I_i \) is
\[
p_i(e_i) = \exp \left( -\frac{d_\cos(e_i, \mu_i)}{2\sigma_i^2} \right),
\]
where \( d_\cos(a, b) = 1 - a^T b \) denotes the cosine distance. Note that \( p_i \) plays both roles of the push and pull terms in the AE loss, as embeddings will be pulled together when the target score increases, and pushed apart otherwise. We use a unit hypersphere to enable the use of cosine distance, which is memory efficient. However, we emphasize that our method is compatible with any distance metric.

**Instance and semantic losses via joint embedding.** For instances, we use a spatial kernel to better separate far away objects as
\[
\psi_i(e_i) = p_i(e_i) \exp \left( -\frac{\|p_i - p_l\|^2}{2\sigma_{l,\text{spatial}}^2} \right),
\]
where \( p_i \) is the spatial position of embedding \( e_i \), and \( \sigma_{l,\text{spatial}} \) is a parameter learned by back propagation.

For handling the learning of semantics, we propose to associate with each semantic class \( S_k \), a semantic mean \( \mu_k \) and semantic variance \( \sigma_k \). We can then write the score of an embedding \( e_i \) belonging to this semantic class via the softmax function
\[
\psi_k(e_i) = \frac{p_k(e_i)}{\sum_{c \in C} p_c(e_i)}. \tag{9}
\]

While it is possible to simply use the Gaussian kernel \( p_k \) for semantics, preliminary experiments indicated that this is not optimal, as unlike instances, the number of semantic classes are known, and thus the decoding can be done by finding the
closest embedding to each semantic mean, which motivates the softmax function.

We define the mean embedding \( \mu_k \) to be the mean embedding inside each image according to the ground-truth for instance indices \( k \) that exist in the current image, and the persistent estimate \( \hat{\mu}_k \) of the dataset mean otherwise:

\[
\mu_k = \begin{cases} 
\frac{1}{|I_k|} \sum_{j \in I_k} e_j & \text{if } k \text{ is an instance} \\
\hat{\mu}_k & \text{if } k \text{ is a semantic class} 
\end{cases} \tag{10}
\]

Given the predicted instance and semantic scores \( \phi_i(e_i) \) and \( \psi_k(e_i) \) for each embedding \( e_i \), we can minimize the Lovász hinge loss, to maximize the IoU metric on the training dataset. Our loss functions that act as the push and pull forces to form the hierarchical structure are defined as

\[
L_{\text{ins}} = \text{LovászBinary}(\phi_i(e_i), t_{\text{ins}}), \tag{11}
\]
\[
L_{\text{seg}} = \text{LovászSoftmax}(\psi_k(e_i), t_{\text{seg}}), \tag{12}
\]

which are calculated for all embeddings \( e_i \), instances \( I_i \) and semantics \( S_k \) and the ground-truth vectors \( t \). This will cause the network to pull embeddings belonging to the same class together, and push apart embeddings that belong to different instances. The instance-semantic hierarchy is modeled by letting the network learn variance estimates that properly capture the hierarchy in the shared embedding space, with the semantic variance being larger than the instance variance.

**Auxiliary losses.** In addition to the embeddings for each pixel, our network also predicts a sigmoid output \( s_i \) for each pixel, estimating the value of the Gaussian kernel \( \phi_i(e_i) \). As shown in previous work, this “seed” value can be used at the decoding step to help finding an arbitrary number of instances [39]. For each pixel, we calculate

\[
L_{\text{seed}} = \frac{1}{|I_i|} \sum_{j \in I_i} \| s_i - \text{sg}([\phi_i(e_i)]) \|^2, \tag{13}
\]

where \( \text{sg}[] \) is the stop gradient operator, e.g. used in VQ-VAE [44]. Note that only instances are predicted to have a seed value, and stuff classes are regressed to zero.

We also add a loss term for each embedding \( e_i \) to encourage the predicted instance variances to be similar for the same object, to ease instance separation:

\[
L_{\text{ins-var}} = \gamma \| \sigma_i - \text{sg}([\sigma_i]) \|^2, \tag{14}
\]

where \( \sigma_i = \frac{1}{|I_i|} \sum_{j \in I_i} \sigma_j \), to encourage uniform predicted variance, and \( \gamma = 10 \).

Similar to previous work by Guerriero et al. [17], we found that the semantic means are difficult to learn via backpropagation, as the update is too slow to catch up with the other parts of the network. Therefore, similar to VQ-VAE [44], for semantic classes, we add an L2 regression loss term to store a persistent representation of each semantic class in the model

\[
L_{\text{seg-mean}} = \| \mu_k - \text{sg}([\mu_k]) \|^2, \tag{15}
\]

where \( \mu_k \) is the batch semantic mean gotten by averaging the predicted embeddings at the ground-truth locations of semantic class \( k \). We illustrate the effect of this VQ-VAE style loss term in our ablation analysis in Section 4.4. The semantic variance \( \sigma_k \) is learned via backpropagation.

**Proposed loss.** Our proposed loss function is independent of the specific network architecture, and can be used with any standard fully convolutional neural network for semantic segmentation, such as DeeplabV3+ [8] or PSPNet [56]. The final form of our proposed loss function is

\[
L = L_{\text{seg}} + L_{\text{seg-mean}} + L_{\text{ins}} + L_{\text{ins-var}} + L_{\text{seed}}, \tag{16}
\]

averaged over all pixels \( X \). Specifically, our network predicts, for each pixel \( x_i \), a tuple \( (e_i, \sigma_i, s_i) \), where \( e_i \) is the predicted embedding, \( \sigma_i \) is the predicted instance variance, and \( s_i \) is the seed probability score.

**Thomson initialization.** To increase coverage of the embedding space, we can initialize the means to uniformly cover the unit hypersphere by solving the generalized Thomson problem [47], originally proposed in 1904 for the purpose of determining an electron configuration in physics. We initialize the means as

\[
\arg\min_{\mu_k \in \mathcal{X}} \sum_{i \neq j} \frac{1}{d_{\text{cos}}(\mu_i, \mu_j)}, \tag{17}
\]

There is no known general solution to this problem, but we can find a local minimum via gradient descent. In our initial experiments, we found that Thomson initialization makes the training more stable.

### 3.5. Panoptic Decoding

For predicting semantics and instances from the learned hierarchical embedding space, we use a simple post-processing algorithm. We first assign the semantic class to each pixel via argmax of \( \psi_k(e_i) \). For thing classes, we first reduce the number of seed candidates \( s_i \) by \( 3 \times 3 \) max pooling, similar to CornerNet [25]. Then we threshold the remaining seeds to get an initial set of candidate seeds. Each candidate seed \( s_i \) has an associated hierarchical embedding \( e_i \), so we can use the kernel \( \Phi(e_i, e_j) = \exp(-d_{\text{cos}}(e_i, e_j)/(2\sigma_i^2) - \| \rho_i - \rho_j \|^2/(2\sigma_i^2)) \) to calculate the probability of two seeds \( (s_i, s_j) \) representing the same instance. We then merge seeds representing the same object based on the probability \( \Phi(e_i, e_j) \). We now have one estimated seed \( s_i \) per instance. For each \( s_i \), we then estimate
<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Pretrain.</th>
<th>(PQ)</th>
<th>(PQ_{th})</th>
<th>(PQ_{st})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposal-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seamless [43]</td>
<td>ResNet50</td>
<td>ImageNet</td>
<td>60.3</td>
<td>56.1</td>
<td>63.3</td>
</tr>
<tr>
<td>UPSNet [52]</td>
<td>ResNet50</td>
<td>ImageNet</td>
<td>59.3</td>
<td>54.6</td>
<td>62.7</td>
</tr>
<tr>
<td>Real-time PS [20]</td>
<td>ResNet50</td>
<td>ImageNet</td>
<td>58.8</td>
<td>52.1</td>
<td>63.7</td>
</tr>
<tr>
<td>Pan.FPN [23]</td>
<td>ResNet50</td>
<td>ImageNet</td>
<td>57.7</td>
<td>51.6</td>
<td>62.2</td>
</tr>
<tr>
<td>Attn.-Guid. [29]</td>
<td>ResNet50</td>
<td>ImageNet</td>
<td>56.4</td>
<td>52.7</td>
<td>59.0</td>
</tr>
<tr>
<td>Li et al. [28]</td>
<td>ResNet101</td>
<td>ImageNet</td>
<td>47.3</td>
<td>39.6</td>
<td>52.9</td>
</tr>
<tr>
<td>DeGeus [13]</td>
<td>ResNet50</td>
<td>ImageNet</td>
<td>45.9</td>
<td>39.2</td>
<td>50.8</td>
</tr>
<tr>
<td>Proposal-free</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSAP [15]</td>
<td>ResNet50</td>
<td>ImageNet</td>
<td>57.6</td>
<td>50.4</td>
<td>-</td>
</tr>
<tr>
<td>DeeperLab [53]</td>
<td>Xception71</td>
<td>ImageNet</td>
<td>56.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DeGeusFast [12]</td>
<td>ResNet50</td>
<td>ImageNet</td>
<td>55.1</td>
<td>48.3</td>
<td>60.1</td>
</tr>
<tr>
<td>HLE (Ours)</td>
<td>ResNet101</td>
<td>ImageNet</td>
<td>60.6</td>
<td>51.4</td>
<td>67.2</td>
</tr>
<tr>
<td>HLE (Ours)</td>
<td>ResNet50</td>
<td>ImageNet</td>
<td>59.8</td>
<td>51.1</td>
<td>66.1</td>
</tr>
<tr>
<td>HLE (Ours)</td>
<td>MobileNetV2</td>
<td>ImageNet</td>
<td>56.0</td>
<td>45.0</td>
<td>64.0</td>
</tr>
<tr>
<td>(PQ)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Single-scale experimental results on the Cityscapes validation set. The best ResNet50 result in each category is highlighted.

The instance mask by thresholding \(\Phi(e_i, e'_i)\), for all remaining pixel embeddings \(e_i\), resolving mask disagreement by assigning each pixel to the instance with the highest probability among the estimated seeds. The semantic class of the instance is decided by the seed pixel. For stuff classes, we threshold the value of the semantic kernel \(\psi_2(e_i)\), to reduce the number of false positives. The panoptic decoding process can be easily optimized in favor of inference speed with little degradation of accuracy (details in the supplementary).

4. Experiments

4.1. Datasets

We conduct experiments on the Cityscapes [10], COCO [30], and Mapillary Vistas [38] datasets, to evaluate the performance of our model. Cityscapes is an image dataset for autonomous driving, depicting European street-level imagery at 1024 \(\times\) 2048 resolution, labeled with 19 classes, out of which 8 are thing classes. COCO contains various indoor and outdoor images at varying resolutions, with 80 thing classes, and 53 stuff classes. Vistas contains a wide variety of street-level images, at varying resolution, with 37 thing classes and 28 stuff classes.

4.2. Experimental Details

We evaluate our method using the standard panoptic segmentation metric panoptic quality (PQ) [24]. In the supplementary material, we explain the \(PQ\) metric, and include its variations, \(PQ^1\) [43] and parsing covering (PC) [53], as well as mean Intersection-over-Union (mIoU) and Average Precision (AP).

On Cityscapes, we only use the fine annotations in training. Our models are trained on the training set (2,975 images) and evaluated on the validation set (1,525 images). We use the Adam optimizer [22] with learning rate 10\(^{-5}\), polynomial learning rate decay, and test-time flip. We use ResNet50 as a backbone for DeepLabV3+ (pretrained on ImageNet) with an embedding dimension of 12 on Cityscapes, and 128 on COCO and Vistas. During training, we use crop size 1024 \(\times\) 2048 on Cityscapes, and 512 \(\times\) 512 on COCO, and crop around thing classes. The experiments on COCO and Vistas uses a max side length of 640 and 2048 pixels, respectively. We use Optuna [1] to find good hyperparameters for the decoder.

Network inference on a 1024 \(\times\) 2048 image takes 91 ms running on single NVIDIA V100 Tensor Core GPU. Post-processing takes 113 ms. Alternatively, by running post-processing on a down-sampled embedding space, post-processing speed can be easily sped up to 8 ms at the expense of lowering Cityscapes validation set PQ to 58.4 (details are provided in the supplementary material). On a 800 \(\times\) 1300 COCO image, inference takes 51 ms. Inference speed depends heavily on the backbone used. With a MobileNetV2 backbone, model inference takes 36 ms on Cityscapes.

4.3. Experimental Results

Our results on Cityscapes, COCO, and Vistas can be seen in Tables 1, 3, and 5, respectively. The best ResNet50
We also report some results of our method with alternative backbones \[\text{ResNet}50\] for previous works, if possible. Depending on the backbone network, our method achieves a fair comparison, we report the results on ImageNet-trained models with a \[\text{ResNet}50\] backbone network beats DeeperLab that uses a much heavier \[\text{Xception}71\] backbone in terms of \(PQ\) metric.

Notably, we achieve a high \(PQS\) score, indicating that using embeddings for semantic encoding is a promising direction, especially in panoptic segmentation. We can reason about the rationale that our model achieves top \(PQS\) results as follows. In our method, we learn both embeddings and variances for each semantic class. Each semantic class is treated in an adaptive manner based on distribution. For example, a larger volume in the embedding space will be assigned to semantic classes with higher variance, compared to direct regression where all classes are treated equally in feature space. An example of the learned hierarchical embedding space can be seen in Figure 3. Our proposed hierarchical embedding is able to encode both semantic and instance level information precisely, although a few mistaken predictions can also be seen.

We also submitted our model to the Cityscapes test set benchmark, where our method is competitive among both published and unpublished models, despite the big variety in network backbones and extra data included in the current benchmark. The results for peer-reviewed work can be seen in Table 2. The gap between our validation and test set \(PQ\) score is not large, indicating that our method generalizes well. While SSAP does not report any \[\text{ResNet}50\] result,

### Table 4. Experimental results on the COCO test-dev2019 set.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Pretrain</th>
<th>(PQ)</th>
<th>(PQ_{ch})</th>
<th>(PQ_{st})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposal-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPSNet [52]</td>
<td>[\text{ResNet}101]</td>
<td>[\text{ImageNet}]</td>
<td>46.6</td>
<td>53.2</td>
<td>36.7</td>
</tr>
<tr>
<td>Attn.-Guid. [29]</td>
<td>[\text{ResNet}Xt-152]</td>
<td>[\text{ImageNet}]</td>
<td>46.5</td>
<td>55.8</td>
<td>32.5</td>
</tr>
<tr>
<td>Pan. FPN [23]</td>
<td>[\text{ResNet}50]</td>
<td>[\text{ImageNet}]</td>
<td>40.9</td>
<td>48.3</td>
<td>29.7</td>
</tr>
<tr>
<td>Proposal-free</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSAP [13]</td>
<td>[\text{ResNet}101]</td>
<td>[\text{ImageNet}]</td>
<td>36.9</td>
<td>40.1</td>
<td>32.0</td>
</tr>
<tr>
<td>DeeperLab [53]</td>
<td>[\text{Xception}71]</td>
<td>[\text{ImageNet}]</td>
<td>34.3</td>
<td>37.5</td>
<td>29.6</td>
</tr>
<tr>
<td>DeeperLab [53]</td>
<td>[\text{Wider MNV2}]</td>
<td>[\text{ImageNet}]</td>
<td>28.1</td>
<td>30.8</td>
<td>24.1</td>
</tr>
<tr>
<td>DeeperLab [53]</td>
<td>[\text{L. W. MNV2}]</td>
<td>[\text{ImageNet}]</td>
<td>24.5</td>
<td>26.9</td>
<td>20.9</td>
</tr>
<tr>
<td>(\text{HLE} \text{ (Ours)})</td>
<td>[\text{ResNet}101]</td>
<td>[\text{ImageNet}]</td>
<td>39.9</td>
<td>45.0</td>
<td>32.2</td>
</tr>
<tr>
<td>(\text{HLE} \text{ (Ours)})</td>
<td>[\text{ResNet}50]</td>
<td>[\text{ImageNet}]</td>
<td>38.2</td>
<td>42.7</td>
<td>31.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Pretrain</th>
<th>(PQ)</th>
<th>(PQ_{ch})</th>
<th>(PQ_{st})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposal-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seamless [43]</td>
<td>[\text{ResNet}50]</td>
<td>[\text{ImageNet}]</td>
<td>37.7</td>
<td>33.8</td>
<td>42.9</td>
</tr>
<tr>
<td>Proposal-free</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pan. DeepL. [9]</td>
<td>[\text{ResNet}50]</td>
<td>[\text{ImageNet}]</td>
<td>33.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DeeperLab [53]</td>
<td>[\text{Xception}71]</td>
<td>[\text{ImageNet}]</td>
<td>32.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(\text{HLE} \text{ (Ours)})</td>
<td>[\text{ResNet}50]</td>
<td>[\text{ImageNet}]</td>
<td>34.6</td>
<td>27.8</td>
<td>43.5</td>
</tr>
</tbody>
</table>

### Table 5. Single-scale experimental results on the Vistas validation set.

The best \[\text{ResNet}50\] result in each category is highlighted.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Pretrain</th>
<th>(PQ)</th>
<th>(PQ_{ch})</th>
<th>(PQ_{st})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposal-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seamless [43]</td>
<td>[\text{ResNet}50]</td>
<td>[\text{ImageNet}]</td>
<td>37.7</td>
<td>33.8</td>
<td>42.9</td>
</tr>
<tr>
<td>Proposal-free</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pan. DeepL. [9]</td>
<td>[\text{ResNet}50]</td>
<td>[\text{ImageNet}]</td>
<td>33.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DeeperLab [53]</td>
<td>[\text{Xception}71]</td>
<td>[\text{ImageNet}]</td>
<td>32.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(\text{HLE} \text{ (Ours)})</td>
<td>[\text{ResNet}50]</td>
<td>[\text{ImageNet}]</td>
<td>34.6</td>
<td>27.8</td>
<td>43.5</td>
</tr>
</tbody>
</table>

### Table 6. Ablative results on the Cityscapes validation set. Our proposed model is in the rightmost column.

<table>
<thead>
<tr>
<th>Component</th>
<th>(\checkmark)</th>
<th>(\checkmark)</th>
<th>(\checkmark)</th>
<th>(\checkmark)</th>
<th>(\checkmark)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assoc. emb. loss</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross-ent. loss</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lovász hinge loss</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sep. seg. branch.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Split emb. space</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hier. emb. space</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VQ-VAE style</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thomson init.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(PQ)</td>
<td>38.8</td>
<td>45.0</td>
<td>50.2</td>
<td>57.0</td>
<td>58.6</td>
</tr>
</tbody>
</table>
our model has higher PQ than theirs with ResNet101. In the supplementary material, we include Cityscapes results comparison on the metrics $PQ^3$ (61.3), $PC^3$ (76.6), $mIoU$ (77.3) and $AP$ (23.9).

On COCO, we saw less proposal-free methods reported with similar network scale as shown in Table 3. We present COCO test set results in Table 4. Our method achieves the best performance among proposal-free models. Compared to Cityscapes, the gap between proposal-based and proposal-free models on COCO is much larger, indicating that there is still a demand for further research in order to close this gap. We find the same observation on the Vistas dataset as on COCO, where our proposed method is able to outperform other proposal-free methods. Notably, we find that bottom-up methods tend to under-perform on larger datasets [9, 20, 15], while our proposed hierarchical embedding space-based method is able to generalize better on these datasets.

4.4. Ablative Analysis

We conduct ablative experiments to better understand design choices we have made for our method. In Table 6, we provide different variations on each key design of our proposed method. For simplicity, we did not conduct hyperparameter search in this experiment.

First of all, we compared our proposed hierarchical Lovász embeddings with conventional embeddings based on associative embedding or cross entropy loss. As can be seen in the table, Lovász embeddings have vastly better performance than associative embeddings or cross entropy losses.

Second, we compare with a model that uses a softmax classifier semantic segmentation branch with cross entropy loss, instead of our joint hierarchical embedding space, and still handling instance predictions via the embedding space. We can see that this yields a lower PQ of 57.0, which can be regarded as a naïve extension of previous instance segmentation work to panoptic segmentation [39]. Next, we illustrate the benefit of jointly learning the embedding space for both semantics and instances, compared to splitting the embedding space into two halves: one for semantics and the other for instance embeddings. We split the 12-dimensional embedding space of our model into 6 dimensions each for semantics and instances. It can be seen that jointly learning both semantics and instances in the same embedding space leads to higher performance. This interpretation is also strengthened by our method outperforming DeeperLab on the Cityscapes validation set – a method using a split embedding space.

Finally, we show that VQ-VAE style loss and Thomson initialization for learning the semantic means help improve the model performance.

4.5. Temporal Stability

As our embeddings are of low dimension, we are interested in seeing whether they exhibit temporal stability when run on a video. In Figure 8, we show the predicted embedding spaces and panoptic segmentation for a few consecutive frames in the Cityscapes demo video sequence. The figure qualitatively shows that matching pixels in different frames have similar embeddings. We believe this qualitative result indicates the potential of our method towards temporal downstream applications such as multi-object tracking. Temporal stability is important for applications such as tracking in autonomous driving, where flickering false positives can be fatal, and low-dimensional embeddings could be leveraged as a feature for cross-frame data association. We are not aware of other panoptic methods featuring this property.

5. Conclusion

In this work, we presented a unified output representation for panoptic segmentation leveraging a hierarchically clustered embedding space. Via our proposed hierarchical Lovász hinge loss, we created a simple single-shot model for panoptic segmentation, predicting an embedding space structured into an instance-semantic hierarchy. Our method was shown to achieve state-of-the-art results among proposal-free panoptic segmentation methods, and be competitive with heavier two-stage methods on the Cityscapes, COCO and Vistas datasets. Results indicated that our hierarchical Lovász embeddings are a viable alternative for panoptic segmentation, for both thing and stuff classes.

We believe that it is an important and promising direction to explore a unified representation for panoptic segmentation. In the future, we will focus on leveraging the structure of the embedding space with self-adaptation to ontology distribution and its downstream applications.

Acknowledgments This work was supported by Woven Core, Inc. We thank Richard Calland, Aditya Ganeshan, Karim Hamzaoui, Daisuke Okanohara and Shunta Saito for discussions and feedback on the manuscript.
References


[38] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In International Conference on Computer Vision (ICCV), 2017. 2, 6
[47] JI Thomson. On the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure. Philos. Mag., Ser. 6, 7:237–265, 1904. 5
