A. Additional Implementation Details

In this section, we describe the implementation details for the proposed PACL method. Particularly, in Appendix A.1, we describe the architecture of the vision embedder used for training PACL, in Appendix A.2, we describe specifics of training including hyperparameters and prompt engineering details. Finally, in Appendix A.3, we describe details of image-text datasets used for training as well as segmentation and image classification datasets used for evaluation.

A.1. Vision Embedder Architecture

In Section 6.1, we have discussed that the proposed PACL approach is flexible in the sense that PACL can be applied using pre-trained frozen encoders. Particularly, since CLIP's pre-trained vision encoders have desirable properties (see Semantic Coherence in Section 4), we use these pre-trained encoders from CLIP to train PACL to transfer to the task of zero-shot semantic segmentation. This simplifies the training to just a small vision embedder on top of the vision encoder. In this section and in Fig. 9, we present the architecture of the Vision embedder. In particular, we use a single residual block with two linear layers in the main branch and a single linear layer in the residual branch. There is a ReLU non-linearity between the two linear layers in the main branch. The resulting model requires training a mere 1.1M parameters whereas the architecture has a total of 150M parameters for CLIP ViT-B/16. This helps us in scaling up and training on a larger batch size for our experiments as there is no gradient propagation through the frozen image and text encoders.

A.2. Training details for Vision Embedder

In Appendix A.2.1, we describe the architecture of pre-trained encoders as well as the hyperparameters used for training the PACL models. In Appendix A.2.2, we provide some details on CLIP's prompt engineering used to derive best results from the text encoder of a pre-trained CLIP model.

A.2.1 Architecture and Hyperparameters

As mentioned above, we only train PACL using a Vision embedder on top of a pre-trained CLIP vision encoder. This allows us the flexibility not only of using multiple pre-trained vision encoders but also combinations of different vision and text encoders. In Section 6.1, we show an ablation with combinations of different pre-trained vision and text encoders. In particular, we use: a) CLIP ViT-B/16 vision and text encoders, b) CLIP ViT-L/14 vision and text encoders and b) DINO ViT-B/16 vision encoder with CLIP ViT-B/16 text encoder. For each of these combinations, we train a vision embedder as discussed in Appendix A.1 and report zero-shot semantic segmentation results in Tab. 3 of the main paper.

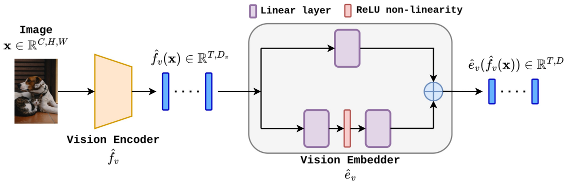
Linear layer
ReLU non-linearity
Image
$\mathbf{x} \in \mathbb{R}^{C,H,W}$
$\hat{f}_v(\mathbf{x}) \in \mathbb{R}^{T,D_v}$
$\hat{e}_v(\hat{f}_v(\mathbf{x})) \in \mathbb{R}^{T,D}$
Vision Encoder
$\hat{f}_v$
Vision Embedder
$\hat{e}_v$

Figure 9. **Vision embedder $\hat{e}_v$ architecture for PACL.** The image encoder $\hat{f}_v$ produces token/patch-wise representations $\hat{f}_v(\mathbf{x}) \in \mathbb{R}^{T,D_v}$ of an input image $\mathbf{x}$. The vision embedder $\hat{e}_v$ converts the patch-wise representations to the multi-modal shared dimensional space, $\hat{e}_v(\hat{f}_v(\mathbf{x})) \in \mathbb{R}^{T,D}$ where T is the number of tokens or patches.

All our models are trained on a single node with 4 NVIDIA A100 GPUs with a GPU memory of 40GB in each GPU. We use AdamW as the optimizer with beta values 0.9 and 0.98, an eps value of $1e-6$ and a weight decay of 0.2. We use an initial learning rate of $5e-4$ and reduce the learning rate using a Cosine Annealing schedule where the maximum number of iterations is set as the total number of iterations during training (i.e., number of epochs × number of iterations per epoch). We use a batch size of 4096 (1024 per GPU) and train the model for a total of 10 epochs on image-text data. *We do not use any segmentation annotations or class-agnostic segmentation masks during training*. We provide details on these image-text datasets in Appendix A.3. When the training dataset is a combination of GCC-3M, GCC-12M and YFCC-15M, the model takes 10 days to train on 4 NVIDIA A100 GPUs.

A.2.2 Prompt Engineering

Since we use CLIP's [41] pre-trained text encoders, we follow the prompt engineering guidelines following CLIP's OpenAI repository during inference time. In particular, during inference, we compute the average embedding from the text encoder using a set of 7 prompts: `itap of a ().`, `a bad photo of the ().`, `a origami ().`, `a photo of the large ().`, `a () in a video game.`,`art of the ()`,`a photo of the small ()`, where we put the name of the class within the parenthesis `()`. We use the mean of the embeddings from the the prompts for each class in order to compute cosine similarity with the patch representations from the vision encoder. This is similar to the way CLIP performs zero-shot image classification, however CLIP uses only the CLS token from the vision encoder to compute cosine similarity.

A.3. Training and Evaluation Datasets

Image-text datasets for training: We use primarily 3 different image-text datasets for training all our models. **Firstly,** we use *Google Conceptual Captions (GCC)*

3M, which contains approximately 3 million images, each annotated with a caption. The images are scraped from the web and the corresponding captions are obtained from the Al-text HTML data associated with each image from the web. **Secondly,** we use *Google Conceptual Captions (GCC) 12M*, which is similar to GCC-3M but containing a much larger corpus of image-text pairs with approximately 12 million samples. The primary purpose of GCC-12M is for pre-training whereas GCC-3M is a relatively less noisy dataset meant for fine-tuning pre-trained models. **Thirdly,** we use *YFCC-15M*, a subset of 15 million samples from the popular YFCC-100M [46] dataset, which is one of the largest publicly available datasets containing image-text information obtained from Flickr. The subset of approximately 15 million images is defined by CLIP [41] by filtering images from YFCC-100M with natural language titles and/or descriptions in English.

Semantic segmentation datasets for zero-shot segmentation: We use the following semantic segmentation datasets for zero-shot evaluation on the task of semantic segmentation: **a)** *Pascal VOC* [16]: it has 20 foreground classes and 1 background class with 1449 validation images. We measure performance only on foreground classes and mask predictions with entropy above 1.5 as background, **b)** *Pascal Context* [36]: it has 59 classes with 5k validation images of indoor and outdoor scenes, **c)** *COCO Stuff* [4]: it has 172 classes categorised into either "thing" classes or "stuff" classes and has 5k validation images, **d)** *ADE20K* [63]: the version we evaluate on is widely used and has 150 classes with 2k validation images. For all datasets, we report the mean intersection over union (mIoU), the most popular evaluation metric for semantic segmentation.

Image classification datasets for zero-shot classification: We evaluate PACL on a suite of 12 image classification datasets which include ImageNet [14], 4 well-known distribution shifts on ImageNet as well as 7 other popular image classification datasets. *ImageNet* is a very popular image classification dataset with 1000 classes relating to concepts contained in the WordNet hierarchy. We use 50000 validation samples in ImageNet for evaluation. The 4 datasets considered to be popular distribution shifts on ImageNet are: **a)** *ImageNet-A* [21] which contains natural real-world images from 200 classes in ImageNet but which are mostly mis-classified by well-known ResNet classifiers, **b)** *ImageNet-R* [20] which contains cartoons, graphics and other art renditions of images from 200 classes in ImageNet, **c)** *ImageNet-Sketch* [48] which contains 50000 validation images, 50 from each of the 1000 ImageNet classes constructed by making the Google search, "sketch of ()" where () is the ImageNet class concerned and **d)** *ImageNet-V2* [42] which has 10000 validation images obtained by following the same collection procedure as ImageNet original images, in order to make the distribution of ImageNet-V2 as similar as possible to ImageNet. The other 7 image classification datasets include: **a)** *CIFAR-10* [27] having 10000 test images from 10 classes including different types of automobiles and animals, **b)** *CIFAR-100* [27] having 10000 test images from 100 classes instead of 10 obtained in a similar fashion as CIFAR-10, **c)** *Stanford Cars* [26] with 8041 test images containing cars of different makes and models, **d)** *Caltech-101* [30] having 101 categories of images with 40-800 images per class, **e)** *Food-101* [2] containing 101 classes of food items organized by the type of food, with approximately 25000 test images, **f)** *Oxford-IIIT Pets* [39], a dataset with 37 categories of pets with approximately 200 images per class, and **g)** *Flower* dataset [37] having 102 different categories of flowers with between 40 and 258 images for each class.

B. Additional Results

B.1. Semantic Coherence in CLIP

Semantic coherence is a property of ViT based vision encoders where semantically similar regions of the image have similar patch/token level representations in the feature space of the vision encoder. In Section 4 and Fig. 4, we have shown both quantitative and qualitative results comparing the semantic coherence of a CLIP and a DINO ViT-B/16 vision encoders. Particularly, we had seen that CLIP's ViT-B/16 vision encoder performs better than DINO. In Fig. 10, we present additional qualitative examples to further corroborate our observations in Section 4. We show qualitative examples from the Bird, Plane and Sheep classes in Pascal VOC and plot the patch level similarity between a selected patch from the original image (marked using a yellow cross) and all patches from the same image as well as a different image. The similarity is shown using a heatmap where yellow and red shades indicate high similarity and blue shades indicate low similarity. Our observations are similar to the ones in Section 4, and we see that CLIP performs competitively or better than DINO. While DINO seems to cover semantically meaningful regions in the images, it doesn't cover the entirety of the relevant object. CLIP seems to be doing a better job at covering all the patches for the object as highly similar to the marked patch, thereby indicating better semantic coherence.

B.2. Qualitative Segmentation Results

In Fig. 6, we showed qualitative results for the task of zero-shot semantic segmentation on both Pascal VOC and ADE20K datasets. In this section, we present more qualitative results on the same. Particularly, in Fig. 11, we show additional qualitative results on 8 images from Pascal VOC covering different concepts including bus, cat, dog, bird, potted plant, bottle and plane. Similarly, in Fig. 12, we pro-

Model	Vision Encoder	Datasets											
		Image Classification								*ImageNet Shifts*			
		ImageNet [14]	*C10* [27]	*C100* [27]	*Cars* [26]	*Caltech101* [30]	*Food101* [2]	*Pets* [39]	*Flowers102* [37]	*ImageNet-A* [21]	*ImageNet-R* [20]	*ImageNet-Sketch* [48]	*ImageNet-V2* [42]
CLIP	ViT-B/16	68.73	91.18	67.88	63.50	85.69	87.52	88.44	61.12	38.88	76.83	48.36	62.21
	ViT-L/14	75.96	95.85	76.94	76.9	86.38	92.69	92.91	69.13	55.44	87.32	59.71	70.26
CLIP + PACL (Ours)	ViT-B/16	73.61	92.3	69.11	60.7	84.8	89.12	90.1	62.3	42.10	78.1	50.14	65.4
	ViT-L/14	78.2	95.13	74.43	74.2	86.25	93.2	93.05	69.7	59.13	85.6	63.23	72.88

Table 6. **Zero-shot Image Classification on 12 different datasets.** We compare PACL's performance with vanilla CLIP for both ViT-B/16 and ViT-L/14 encoders. The first 8 datasets are standard image classification datasets: ImageNet, CIFAR-10, CIFAR-100, Stanford Cars, Caltech101, Food101, OxfordIIITPets, and Flowers102. The remaining 4 datasets are standard distribution shifts on ImageNet: ImageNet-A, ImageNet-R, ImageNet-Sketch and ImageNet-V2. PACL + CLIP broadly outperforms vanilla CLIP on most of the classification datasets.

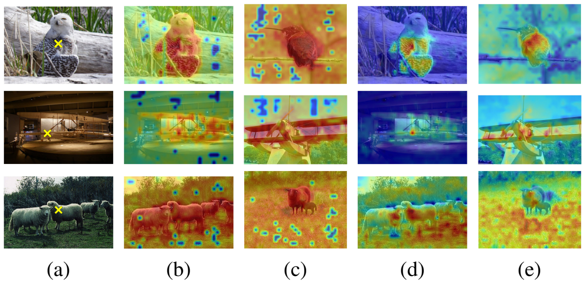

Figure 10. **Additional qualitative results on semantic coherence between CLIP and DINO ViT-B/16. a)**: we show the original image of a class (bird in top row, aeroplane in middle row and sheep in bottom row) with the patch marker (yellow X near the centre). **b, c)**: we show CLIP vision encoder cosine similarity across all patches for the same and a different image of the same class. **d, e)**: we show the same for DINO.

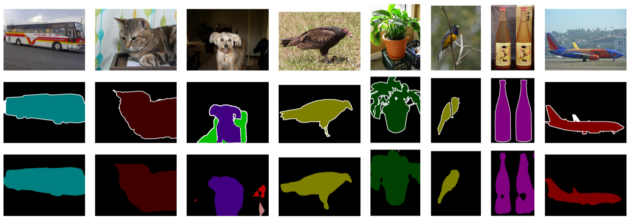

Figure 11. **Additional qualitative results on zero-shot semantic segmentation on Pascal VOC.** The first row shows the original images, the second row shows the corresponding ground-truth labels and the third row shows the predictions from our best performing model, i.e., PACL trained using a pre-trained CLIP ViT-B/16 encoder on GCC-3M + GCC-12M + YFCC-15M.

vide qualitative segmentation results from various indoor and outdoor scenes from ADE20K. Similar to our observations in the main paper, we find that the zero-shot segmentation results are decent and our models can recognise a large variety of concepts without ever having been trained on segmentation annotations or masks for any of them. This shows the potential of using the scale of large image-text datasets for zero-shot transfer to semantic segmentation.

B.3. Zero-shot Image Classification

In Fig. 8 of the main paper, we showed the difference in zero-shot classification accuracies for PACL models trained with CLIP backbones as compared to vanilla CLIP models on a suite of 12 image classification tasks including ImageNet, 4 datasets considered to be distribution shifts on ImageNet and 7 other well-known image classification datasets. In this section, we provide the exact classification accuracies for all the models on each of the datasets. We present these results in Tab. 6. As mentioned in the main paper, the PACL models outperform vanilla CLIP on 10 out of 12 datasets for the ViT-B/16 model and 7 out of 12 datasets for the ViT-L/14 backbone, thereby broadly outperforming CLIP on zero-shot image classification.

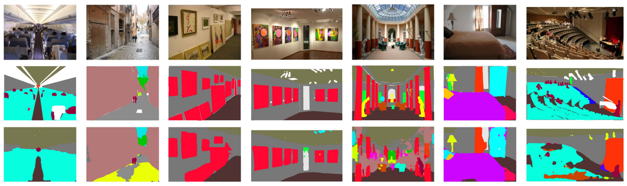

Figure 12. **Additional qualitative results on zero-shot semantic segmentation on ADE-20K.** The first row shows the original images, the second row shows the corresponding ground-truth labels and the third row shows the predictions from our best performing model, i.e., PACL trained using a pre-trained CLIP ViT-B/16 encoder on GCC-3M + GCC-12M + YFCC-15M.

C. Future Work

In this section, we discuss possible avenues for future research based on our work.

Exploring PACL for image-level applications: As seen above, since PACL is a general compatibility function for contrastive loss, it can be applied to all image level tasks. We show this through zero-shot image classification. However, it would be interesting to further explore PACL as an independent contrastive learning method. In particular, training models from scratch on PACL instead of the standard CLIP loss might provide additional benefits in the context of general VLP tasks like image-text retrieval [50]. Since our work is focused around zero-shot semantic segmentation, we keep this exploration out of the scope of this work and as potential avenue for future research.

Exploring other ways to generate patch level alignment: All the current methods on zero-shot open vocabulary segmentation, including ours, use CLIP like models, i.e., models with individual vision and text encoders with a fusion of modalities at the end of the encoders. However, there could be other ways of fusing modalities which could also lead to a generation of patch level alignment between image and text. In particular, one of the seemingly likely candidates of multi-modal fusion for generating patch level alignment could be cross-attention between image and text tokens, often seen in architectures used in VLP training [25, 31, 45] etc. Studying the patch level alignment in these models to see if they can be transferred to dense prediction tasks is also an interesting area of future exploration.