AdaSemSeg: An Adaptive Few-shot Semantic Segmentation of Seismic Facies

Surojit Saha[1] and Ross Whitaker[1]

Scientific Computing and Imaging Institute, Kahlert School of Computing, The University of Utah, Salt Lake City, USA {surojit.saha,ross.whitaker}@utah.edu

Abstract. Automated interpretation of seismic images using deep learning methods is challenging because of the limited availability of training data. Few-shot learning is a suitable learning paradigm in such scenarios due to its ability to adapt to a new task with limited supervision (small training budget). Existing few-shot semantic segmentation (FSSS) methods fix the number of target classes. Therefore, they do not support joint training on multiple datasets varying in the number of classes. In the context of the interpretation of seismic facies, fixing the number of target classes inhibits the generalization capability of a model trained on one facies dataset to another, which is likely to have a different number of facies. To address this shortcoming, we propose a few-shot semantic segmentation method for interpreting seismic facies that can adapt to the varying number of facies across the dataset, dubbed the *AdaSemSeg*. In general, FSSS methods use the statistics of the pretrained backbone network learned from the ImageNet dataset to make predictions on unseen target classes without any refinement on the target dataset. The lack of such a dataset for seismic images motivates using a self-supervised algorithm on seismic datasets to initialize the backbone network. We have trained the AdaSemSeg on three public seismic facies datasets with different numbers of facies and evaluated them on multiple metrics. The performance of the AdaSemSeg on *unseen* datasets is better than the prototype-based few-shot method and comparable to the baselines.

Keywords: Few-shot semantic segmentation · Seismic facies interpretation · Self-supervised learning.

1 Introduction

The study of facies in seismic images has emerged as an important topic of research in the recent past due to the comprehensive characterization of the earth's subsurface. Seismic facies represent regions in the earth's crust with similar geological characteristics, as indicated by correlated reflection properties in seismic images. In addition to identifying hydrocarbon reservoirs, delineating horizons in seismic images into facies find potential applications in capturing and storing carbon dioxide. Manual interpretation of seismic facies by expert geologists is a painstacking process. Moreover, the presence of complex morphological variations often leads to subjective interpretation. Thus, automated detection is encouraged for efficient solutions that also alleviate the biased interpretation.

Advancements in deep learning methods have accelerated the automated interpretation of seismic images [1, 26]. Deep learning techniques produce impressive results with a sufficiently large training dataset. However, obtaining a large annotated seismic dataset is costly and impractical in most scenarios. Therefore, a potential approach would be the induction of a learning paradigm that can quickly adapt to the target task while having limited supervision. This learning technique is known as the *few-shot* learning. The few-shot learning involves two stages: the *meta-training* and *meta-testing*. In the meta-training stage, large annotated data related to but different from the target task is used to train a deep neural network. The meta-testing stage deals with adapting the model to the new task *with/without* fine-tuning the meta-trained parameters. Broadly, the few-shot learning algorithms are categorized as model-based (black-box) [17], metric-based (non-parametric) [24] or optimization-based methods [9].

Few-shot semantic segmentation (FSSS) methods do multi-class classification at the pixel level, an extension to the binary segmentation task. Generally, the FSSS methods need to know *a priori* the number of classes present in the dataset, which limits the use of the method on multiple datasets varying in the number of target classes. The community has recognized this limitation, and methods have been built in the recent past to alleviate this issue [14, 29, 10]. The MSeg [14] and LMSEG [29] build a unified taxonomy of classes present in multiple semantic segmentation datasets by interpreting the names of the class labels. The viability of constructing a unified taxonomy for interpreting seismic facies is questionable due to the variability in the geological features observed across datasets. This motivated us to devise a generalized few-shot segmentation method that is independent of the facies' names and can accommodate the variability in the number of facies across multiple datasets.

We propose a FSSS method for identifying seismic facies using Gaussian processes that can adapt to different numbers of facies across datasets, named the AdaSemSeg. The proposed technique is motivated and adapted from the method introduced by Johnander *et al.* [12] that empirically demonstrated the strength of the Gaussian process regression in the few-shot setup. In this work, we devised a technique that extends the capability of the method in [12] (designed for binary segmentation) to do multi-class segmentation (aka semantic segmentation). In the AdaSemSeg, the multi-class segmentation problem is divided into multiple binary segmentation tasks, where each task recognizes a particular type of facies in a given dataset. An individual task predicts a type of facies in a *query* image using the statistics of the posterior distribution estimated by the Gaussian process regression from the *support* images. The number of binary tasks for a dataset depends on the number of facies present. The final multi-class prediction for a query image is obtained by aggregating the outcomes of multiple binary segmentation tasks. We could successfully train the AdaSemSeg on three public datasets with different facies. Besides the semantic segmentation of facies, the proposed model can be trained jointly to recognize other geological features, such as faults and salt domes.

FSSS methods use the statistics of the ImageNet dataset [8] to initialize the backbone network, such as the ResNet [11], to make predictions on unseen target classes *without refinement* on the target dataset. However, we do not have any benchmark dataset similar to ImageNet for seismic image analysis. We observed that the performance of the few-shot segmentation methods [25, 12] is affected by the initialization of the network parameters. Besides the large annotated datasets [8], self-supervised learning offers an alternative approach to initializing the backbone image encoder. In this work, we use contrastive self-supervised learning [5] that uses positive and negative examples produced from samples in a dataset to train the network parameters.

The AdaSemSeg has been jointly trained on three benchmark 3D facies datasets, the F3 from the Netherlands [1], Penobscot from Canada [2] and Parihaka from New Zealand [4] having 6, 7 and 6 different facies, respectively. We compare the performance of the AdaSemSeg (*without refinement*) with different baselines that are trained on the target facies dataset using multiple evaluation metrics used in the literature [1, 26]. In addition, the performance of the AdaSemSeg is compared with a prototype-based few-shot segmentation method developed for seismic facies [28] and a regular semantic segmentation network *fine-tuned* on the target dataset (aka *transfer learning*). A summary of the contributions of this work is as follows:

- We propose an adaptive FSSS method for identifying seismic facies that is flexible to handle the variability in the number of facies across datasets.
- Initialization of the image encoder using contrastive learning helps improve the AdaSemSeg's performance.
- The performance of the AdaSemSeg evaluated on target datasets *without* refinement is comparable to the baselines.
- The AdaSemSeg comprehensively outperforms the prototype-based FSSS method and the segmentation model trained using transfer learning.

2 Related Work

Deep learning methods have been used to recognize geological features, such as faults and channels, in seismic images [7, 20]. With the release of multiple seismic facies datasets in the recent past [1, 2, 4], deep learning algorithms have been proposed for the identification of seismic facies [26, 3, 13]. Among the public datasets, the data released as a part of a SEG challenge on the Parihaka 3D volume [3] has gained much attention due to its complexity. Different segmentation networks, such as the DeconvNet [1], U-Net [26, 3], and DeepLabv3 [13] have been used for the interpretation of facies.

All the segmentation methods developed for the identification of seismic facies [1, 6, 26, 3, 13] rely on a large set of annotated data, which does not present a realistic scenario as the production and annotation of seismic images are costly enterprise. In reality, we would use a pretrained model (trained on benchmark datasets) to make predictions on a new dataset with a handful of annotations. Few-shot segmentation (FSS) is an effective technique used to address these

scenario [22, 25, 12], which has been extended to multi-class classification [27, 23]. A couple of FSSS methods have been developed for the segmentation of seismic facies [15, 28]. The method proposed in [28] is the closest to our approach as it deals with multiple datasets varying in the number of classes.

Among the different FSS methods, we find the DGPNet [12] using Gaussian processes (GP) in the latent space to be an effective method to be used for the segmentation of seismic facies. The GP is a probabilistic regression technique that can quickly adapt to the observed data. The DGPNet uses GPs to predict the segmentation mask for the query image using the support examples. The DGPNet has three trainable networks, the image backbone (F), mask encoder (G), and decoder (U), which are trained jointly in an end-to-end setup using stochastic gradient descent. The decoder combines the predicted masks (obtained from GPs) at multiple coarse layers and the encoded image features (at shallow layers) to predict the mask on the query image at the pixel space. The use of the DGPNet for the interpretation of biomedical images [19] demonstrates the method's efficacy beyond natural images.

3 Method

3.1 Background

Few-shot Segmentation: Few-shot learning is a meta-learning algorithm [9, 24] that aims to generalize the behavior of a model trained on *source* classes, having a lot of labeled examples, to *unseen/novel* classes (known as the *target* classes) that have a few annotated samples. FSS methods [22] predict the binary mask for the *unseen* class in an input image, known as the *query* image, using a handful of annotated samples from the target class, known as the *support* set. We denote the source dataset containing N different classes as $\mathcal{X}_s = \{\mathcal{X}_s^1, .., \mathcal{X}_s^N\}$, where $\mathcal{X}_s^c = \{(I, M)^i\}_{i=1}^{|\mathcal{X}_s^c|}$ are the annotated examples for the source class c. Similarly, the target data set containing M different classes are defined as $\mathcal{X}_t = \{\mathcal{X}_t^1, .., \mathcal{X}_t^M\}$, where $\mathcal{X}_t^d = \{(I, M)^i\}_{i=1}^{|\mathcal{X}_t^d|}$ are the annotated examples for the target class d.

The FSS is an episodic learning algorithm that follows the N-way K-shot structure to define a task used to train the model parameters, where N and K represent the number of classes and examples (or shots) from each class, respectively. As reported in the literature [22, 12], we use $N = 1$ to define a task, i.e., we select a class at random and sample K examples from it as the *support* set and a single example from the same class as the *query* sample. In this work, we denote the *support* set as $S = \{(I^s, M^s)^i\}_{i=1}^K$ and the query set as $Q = \{(I^q, M^q)\}$, where $I \in R^{H \times W \times 3}$ is the *support/query* image and $M \in \{0, 1\}^{H \times W}$ is the corresponding mask.

In this learning framework, multiple tasks (a combination of support and query images) are produced from the source classes to train the model. The model predicts the mask, $\hat{M}^q$, on the query image, I^q, using the samples in the support set, S, which is compared with the ground truth M^q to minimize the

mismatch. Under this setting, we train the segmentation model end-to-end using stochastic gradient descent.

3.2 The Proposed Method: AdaSemSeg

The AdaSemSeg uses class-specific predictions from the binary segmentation module to recognize facies, a multi-class classification problem, on the query image. The binary segmentation module in the AdaSemSeg does a GP regression in the latent space to predict a particular category of facies, similar to the binary segmentation in the DGPNet [12]. This process is repeated for all the types of facies in a dataset, resulting in multiple GP regressions, a key difference with the DGPNet. The aggregation of predictions from the shared binary segmentation module to do semantic segmentation makes the AdaSemSeg adaptive to the number of facies present in a dataset. Thus, we do not need information about the number of facies present in a dataset to design our model, which other standard segmentation methods use. The proposed approach allows us to train the AdaSemSeg on multiple datasets varying in the number of facies.

For the binary segmentation task in the AdaSemSeg, the original labeled data, M^s, having multi-class annotations is processed to produce binary segmentation masks, M^s_c, representing the mask for class c, as shown in Fig. 1(a) for an example in the support set. The number of binary masks produced for a labeled input depends on the number of facies, C, in the dataset, resulting in $M^{s_i} = \{M^{s_i}_{c_j}\}^C_{j=1}$ for a labeled mask M^{s_i} in the support set. We produce *class-specific annotations* using this processing step, for the support (S) and query (Q) set, defined as $M^S_{c_j} = \{M^{s_i}_{c_j}\}^K_{i=1}$ and $M^Q_{c_j} = \{M^q_{c_j}\}$, respectively, for class c_j. Fig. 1(b) shows the class-specific labels, $\{M^S_{c_1}, M^Q_{c_1}\}$ and $\{M^S_{c_6}, M^Q_{c_6}\}$ for the support and query set, representing class c_1 and c_6 of the F3 facies dataset, respectively. Processing of the input using this technique results in multiple class-specific annotations, depending on the number of facies (C), for the same images (I) in the support and query set.

The AdaSemSeg has three trainable modules, namely, the image encoder (IE) that encodes images in the support and query set, the mask encoder (ME) that encodes the class-specific binary masks in the support set and, the decoder (D) that processes the output of the GP regressions in the latent space and the shallow encoded image features to predict binary masks, as shown in Fig. 2. Using the support images, I^S, and class-specific binary masks for the class c_j, $M^S_{c_j}$, in the support set, the segmentation network predicts a binary mask, $\hat{M}^q_{c_j}$, on the query image, using the prediction of the GP regression (in deep layers) and the encoded query image features (in the shallow layers). The GP in a latent layer in the AdaSemSeg learns the mapping from the encoded image space, R^F, to the encoded mask space, $R^{F'}$, and statistics of the class-wise posterior distribution, $P_{c_j} = \{\mu_{q_{c_j}|S_{c_j}}, \Sigma_{q_{c_j}|S_{c_j}}\}$, estimated by the regression model [16] are defined as,

$$\mu_{q_{c_j}|S_{c_j}} = K^T_{Sq}(K_{SS} + \sigma^2_z\mathbf{I})^{-1}e^{S'}_{c_j} \in R^{H'\times W'\times F'} \tag{1}$$

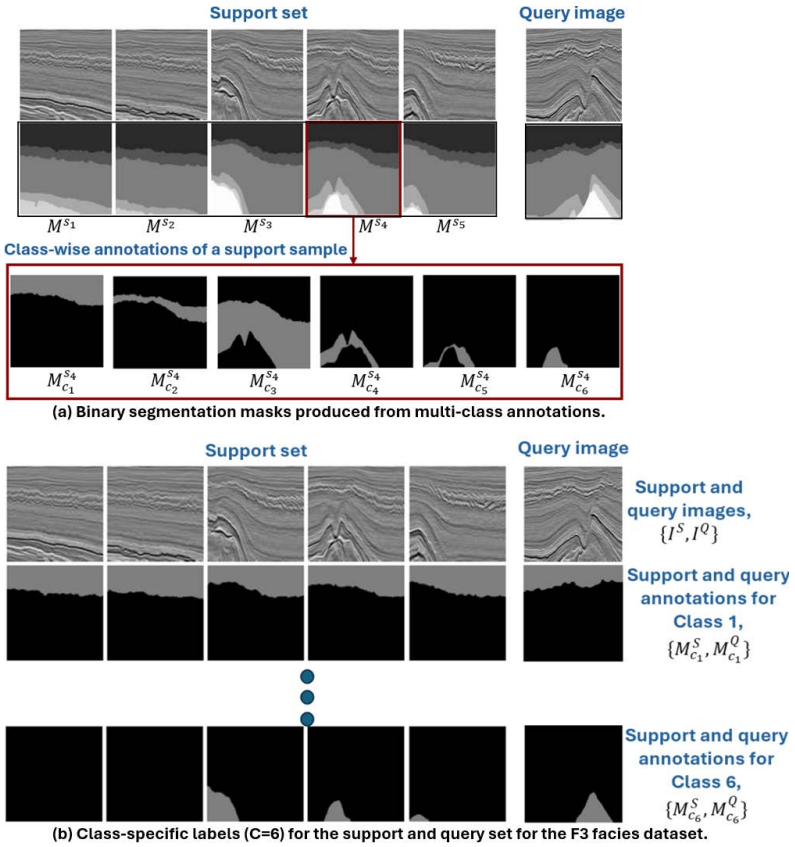


Fig. 1. The processing of the input to produce class-specific annotations on the support and query images. (a) Processing of the multi-class labeled example, M^{s_4}, in the support set to produce binary segmentation masks for 6 classes in the F3 dataset [1]. (b) Class-specific labels for the support (S) and query set (Q) for class c_1 and class c_6 of the F3 facies dataset [1].

$$\Sigma_{q_{c_j}|S_{c_j}} = K_{qq} - K_{Sq}^T(K_{SS} + \sigma_z^2\mathbf{I})^{-1}K_{Sq} \in R^{H^{'}W^{'}\times H^{'}W^{'}}, \tag{2}$$

where K_{SS}, K_{qq}, and K_{Sq} are the co-variance matrices computed using the encoded support images, $e^S = \{e^{s_i}\}_{i=1}^K$, and encoded query image, e^q. The encoded representation of a *support* image, I^s, and its corresponding mask for class c_j, $M^s_{c_j}$, are denoted as $e^s = IE(I^s) \in R^{H^{'}\times W^{'}\times F}$ and $e^{s^{'}} = ME(M^s_{c_j}) \in R^{H^{'}\times W^{'}\times F^{'}}$, respectively. $KH^{'}W^{'}$ image and mask encodings are produced from K samples in the class-specific support set, $S_{c_j} = \{(I^s, M^s_{c_j})^i\}_{i=1}^K$, in the F and $F^{'}$ dimensional feature space, respectively. Similarly, the encoded representation of a *query* image, I^q, defined as $e^q = IE(I^q) \in R^{H^{'}\times W^{'}\times F}$ results in $H^{'}W^{'}$ examples in the encoded image space. The noise in the labeled data is represented

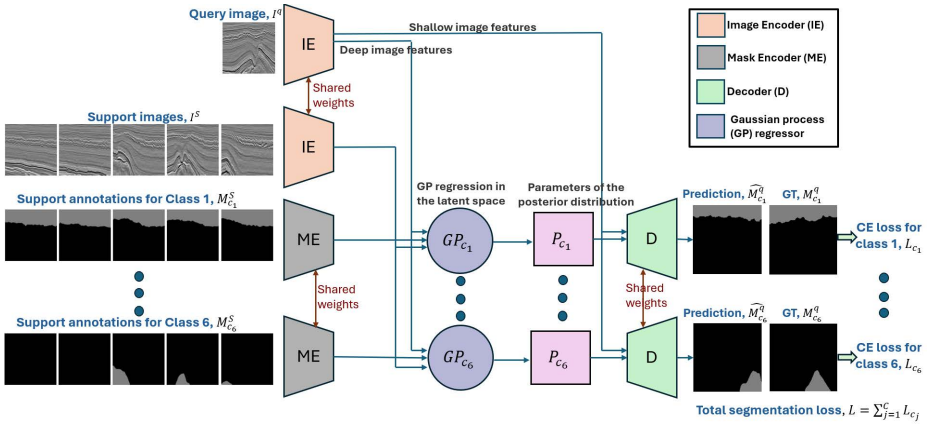


Fig. 2. The neural network architecture of the AdaSemSeg and its training on the F3 facies dataset [1], which extends to other datasets without any modifications.

by σ_z. A squared exponential kernel is used for computing the co-variance matrices, K_{SS}, K_{qq}, and K_{Sq}. The GP regression is used in two latent spaces of the AdaSemSeg for each class, c_j, the bottleneck layer, and the layer above it. Under this setting, a GP regression is performed separately on each class in the facies dataset using the class-specific annotations on the support and query set, $M^S_{c_j}$ and $M^Q_{c_j}$, respectively. In Fig. 2, the GP regression uses the class-specific support sets, S_{c_j}, to obtain predictions on the query image, I^q, separately for each class in the F3 facies dataset [1]. As there are $C = 6$ classes in the F3 facies dataset, the AdaSemSeg predicts the masks individually on all the classes using the class-specific labels, as shown in Fig. 2. This illustration elucidates the use of multiple GPs for making predictions on a query image.

The predicted mask on the query image for class c_j, $\hat{M}^q_{c_j}$, is compared with the ground truth annotation, $M^q_{c_j}$ to compute the loss for the class c_j, L_{c_j}. We perform class-wise predictions for all the classes, C, in the facies dataset using the shared segmentation network (i.e., the IE, ME, and D) and the class-specific binary masks. The AdaSemSeg accumulates the loss across all the classes, named as the *total segmentation loss*, $L = \sum_{j=1}^{C} L_{c_j}$. We use the pixel-wise cross-entropy loss to compare the predicted masks with the ground truth, and the segmentation loss across multiple classes is defined as,

$$L = -\frac{1}{HW} \sum_{j=1}^{C} \sum_{h=1}^{H} \sum_{w=1}^{W} M^q_{c_j}(h,w) \log \hat{M}^q_{c_j}(h,w). \tag{3}$$

The loss computed on a minibatch is used to update the shared trainable parameters using stochastic gradient descent. The predictions on different facies types in a dataset are combined using a Softmax operation to obtain the multi-class prediction on the query image.

Initialization of the image encoder: In general, few-shot methods rely on backbone networks (e.g., ResNet [11]) as image encoders that have been pretrained on the ImageNet data set [8], which is not only very large but also entails recognition of natural objects that are associated with the segmentation of target classes (e.g., bicycles, cats). However, we do not have access to such a vast public dataset in seismic image interpretation. Potential reasons are the challenges associated with seismic imaging, annotation of seismic images requiring specially skilled labor (unlike natural images), and the cost involved in the process. We have observed that the initialization strategy significantly affects the performance of the FSS and FSSS methods developed for natural images. Thus, having a pretrained backbone network for seismic datasets is important. This motivated us to use the *self-supervised* learning technique to initialize the image encoder with the statistics of the seismic image datasets. In this work, we use the SimCLR [5], a contrastive self-supervised algorithm for learning representations to assist the segmentation of seismic facies. The details of the experimental setup, such as the augmentations used in generating positive and negative pairs, batch size, and other optimization configurations, are discussed in section 4.1.

4 Experiments

4.1 Experimental Setup

Datasets: We use the Penobscot [2], Parihaka [4], and F3 [1] facies datasets for evaluation of the AdaSemSeg under different scenarios. The train, validation, and test set distribution for all the datasets are shown in Fig. 3. For the F3 dataset, we use the train-val-test split proposed in [1]. The Penobscot dataset is processed to remove the corrupted images that resulted in 460 inline and 471 crossline images from the original volume containing 601 inline and 481 crossline slices [6]. With the expert's suggestion, we reduced the depth of the Penobscot dataset to 900 (originally 1501). We do not merge underrepresented horizons, such as layers 2 and 3, as done in [6]. The Parihaka facies dataset was released as a part of the challenge, where the organizers withheld the original test annotations. Thus, we select a small part of the training data as the test data for evaluating different methods, such that the selected test slices are close to the real test data along the inline and crossline directions [3]. We process the Penobscot and Parihaka datasets using percentile-based filtering, such as the $1-99$ percentile, to remove the extreme intensity values (aka *outliers*). We estimate an acceptable range of values [*lower-threshold*, *upper-threshold*] using the percentile-based filtering, such that intensity values less than the *lower-threshold* are clipped to *lower-threshold*, and values higher than the *upper-threshold* are mapped to *upper-threshold*. This work uses the $5-95$ percentile range to remove the outliers. We re-scale the values of all the datasets to the range of $0-255$. 2D patches of size 256×256 are extracted along the inline and crossline directions to train the AdaSemSeg and other methods studied in this work for all the data volumes [1, 26, 28]. Using whole slices instead of patches gives better results for a dataset as demonstrated

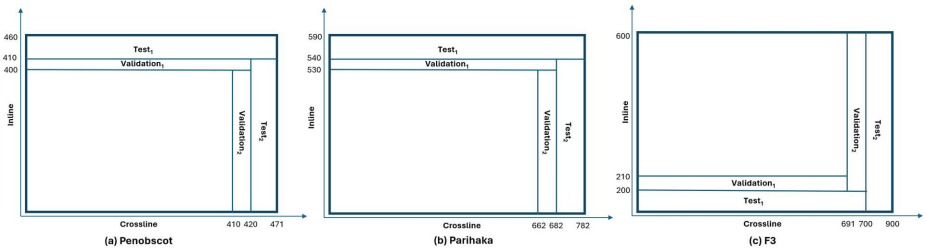


Fig. 3. The train, validation and test data distribution of the (a) Penobscot [2], (b) Parihaka [4] and (c) F3 [1] facies datasets along the inline and crossline directions.

in [1, 3]. However, we hypothesize that patches offer more variability in the training data, resulting in better generalization to other unseen datasets. In addition, whole slices require a lot of GPU memory to train the deep neural networks, which can be impractical for large data volumes. We use the leave-one-out policy to create the data for the *meta-training* and *meta-testing*. For example, to evaluate the AdaSemSeg on the Parihaka dataset (target dataset used in the meta-testing), we train the AdaSemSeg on the Penobscot and F3 dataset in a few-shot setup (source datasets in the meta-training). Similarly, the AdaSemSeg is evaluated on the Penobscot and F3 datasets when the model is trained on the remaining two datasets. Under this experimental setting, we evaluate the generalization of the AdaSemSeg to unseen target datasets.

Baseline and other competing methods: The AdaSemSeg is evaluated on the target dataset *without refinement* of the network parameters on the target dataset. This evaluation setup is consistent with other FSSS methods that predict the segmentation masks on unseen target classes. We choose the AdaSemSeg trained *only* on the target dataset as one of the baseline, Baseline-1, and a U-Net-based [18] segmentation network, named the Baseline-2, as another baseline, which is also trained on the target dataset. Besides, we compare the performance with a prototype-based [24] few-shot segmentation method for seismic facies [28], referred to herein as the *ProtoSemSeg*. Transfer learning has been widely used in deep learning research, where the last few layers of a deep neural network are retrained (sometimes with modifications) with few examples from the target task. It is used in the interpretation of seismic images, such as the classification of seismic facies [21] and detection of faults [7]. We train the Baseline-2 using transfer learning, where the model is trained using patches from the source data, e.g., the F3 and Penobscot volume and fine-tuned on a handful of annotated data from the Parihaka volume, the target class. The last layer of the decoder in the Baseline-2 is customized to the number of classes in the facies dataset, such as 7 for the Penobscot and 6 in the case of the Parihaka and F3 datasets.

Evaluation metrics: The performance of the AdaSemSeg and other methods is evaluated under different metrics used in the literature [1, 26], such as the pixel

Table 1. Evaluation of the AdaSemSeg on the F3, Penobscot, and Parihaka datasets using $k = 5$ support examples or the *nearest slice* in the support set. The use of the $k = 5$ support examples for evaluating the query image is indicated by ✗ and the ✓ represents the use of the *nearest slice* as the support example. The best performance of the AdaSemSeg under different evaluation scenarios is highlighted in **bold**.

Dataset	Nearest slice	inline				Nearest slice	crossline			
		PA	MCA	FwIoU	FwF_1		PA	MCA	FwIoU	FwF_1
F3	✗	**0.89**	**0.79**	**0.81**	**0.89**	✗	**0.87**	**0.73**	**0.80**	**0.88**
	✓	0.85	0.73	0.78	0.85	✓	0.80	0.58	0.71	0.81
Penobscot	✗	0.95	**0.95**	0.91	0.96	✗	**0.97**	**0.95**	**0.93**	**0.96**
	✓	**0.96**	**0.95**	**0.94**	**0.97**	✓	0.96	0.94	0.92	0.95
Parihaka	✗	0.78	0.68	0.66	0.79	✗	0.79	0.65	0.67	0.80
	✓	**0.86**	**0.76**	**0.76**	**0.86**	✓	**0.84**	**0.68**	**0.74**	**0.85**

accuracy (PA), class accuracy, mean class accuracy (MCA), intersection over union (IoU), and F_1 score. The IoU and F_1 scores are weighted by the frequency of the classes, denoted by the FwIoU and FwF_1 scores, respectively. We use whole slices for evaluation of all the methods (supported by the CNNs) for practical use cases and to avoid the complexity of patch stitching [26]. Moreover, patch stitching does not offer any additional performance benefits.

4.2 Experimental Results

The image encoder, a ResNet50, is trained on the patches extracted from the F3, Penobscot, and Parihaka datasets using the SimCLR algorithm [5]. The AdaSemSeg uses the image encoder trained using the SimCLR algorithm. For all the results discussed in this section, we assume that after completing the meta-training on source datasets, the AdaSemSeg is evaluated on the target dataset *without refinement* of the network parameters on the target dataset unless mentioned explicitly. The details of the image encoder (IE), and decoder (D) used in the AdaSemSeg, Baseline-2 and ProSemSeg along with the mask encoder (ME) used by the AdaSemSeg are reported in the Appendix. Refer to the Appendix for the configuration of the optimizer, learning rate scheduler, augmentations and other experimental details.

The use of the support examples for evaluation: The AdaSemSeg uses $k = \{1, 5\}$ examples from the training volume of the target dataset as the support set (S) that spans the whole volume to predict the facies on the query images, I^qs, which are samples in the test set of the target dataset. For example, the indices of the support set along the crossline direction for the F3 facies dataset are $S_i = \{0, 171, 343, 514, 686\}$ and the test data starts at the index 700 (refer to Fig. 3 for details). Instead of k support images, we can use a single sample in the support set closest to the query slice as the support sample, referred to herein as the *nearest slice*. Under this evaluation scenario, we use the slice at index $686 \in S_i$ as the support set (S) to predict the mask for the same test data. We

Table 2. Comparison of the AdaSemSeg (w/o refinement) with baselines trained *only* on the target datasets. NA under several class indices for different datasets indicates the absence of the corresponding class indices in the ground truth annotations. The performance of the best method is highlighted in **bold**.

Target Dataset	Method	PA	Class accuracy							MCA	FwIoU	FwF_1
			1	2	3	4	5	6	7			
Parihaka inline	AdaSemSeg	0.86	0.98	0.87	0.37	0.80	NA	0.76	NA	0.76	0.76	0.86
	Baseline−1	**0.91**	**0.99**	**0.92**	0.75	0.83	NA	**0.87**	NA	**0.87**	**0.83**	**0.90**
	Baseline−2	0.87	0.85	0.87	**0.76**	**0.88**	NA	**0.87**	NA	0.85	0.77	0.86
Parihaka crossline	AdaSemSeg	0.84	**0.84**	**0.91**	0.63	0.82	0.02	0.83	NA	0.68	0.74	0.85
	Baseline−1	**0.91**	0.83	**0.91**	0.89	0.90	**0.82**	**0.97**	NA	**0.89**	**0.85**	**0.92**
	Baseline−2	0.86	0.76	0.88	**0.92**	**0.93**	0.65	0.77	NA	0.82	0.78	0.88
Penobscot inline	AdaSemSeg	0.95	0.97	0.99	0.92	0.94	0.99	0.90	0.91	0.95	0.91	0.96
	Baseline−1	**0.97**	**0.99**	**1.00**	**0.99**	0.98	0.98	**1.00**	**0.92**	**0.98**	**0.95**	**0.98**
	Baseline−2	**0.97**	**0.99**	0.99	**0.99**	**0.99**	**1.00**	0.99	0.90	**0.98**	**0.95**	**0.98**
Penobscot crossline	AdaSemSeg	0.97	0.96	0.97	0.92	0.88	0.98	0.96	0.98	0.95	0.93	0.96
	Baseline−1	0.97	**0.98**	**0.98**	**0.99**	0.94	0.96	**0.99**	**0.97**	0.97	0.94	**0.97**
	Baseline−2	**0.98**	**0.98**	0.97	0.98	**0.97**	**0.99**	**0.99**	**0.97**	**0.98**	**0.95**	**0.97**
F3 inline	AdaSemSeg	0.89	0.97	**0.94**	0.96	0.76	0.66	0.43	NA	0.79	0.81	0.89
	Baseline−1	**0.91**	**0.99**	**0.94**	**0.97**	0.81	**0.73**	**0.88**	NA	**0.89**	**0.86**	**0.91**
	Baseline−2	0.88	**0.99**	**0.94**	0.96	**0.89**	0.56	0.67	NA	0.84	0.79	0.87
F3 crossline	AdaSemSeg	0.87	0.94	**0.89**	**0.88**	0.68	0.18	0.80	NA	0.73	0.80	0.88
	Baseline−1	0.86	**0.98**	0.86	0.85	0.61	0.17	**0.98**	NA	0.74	0.80	0.87
	Baseline−2	**0.89**	**0.98**	0.85	0.87	**0.82**	**0.36**	0.88	NA	**0.79**	**0.82**	**0.89**

hypothesize that the second approach to constructing the support set is likely more effective when significant morphological changes are present along an axis, as observed in the Parihaka dataset (refer to the Appendix for visualization).

We evaluate the AdaSemSeg using the sampling techniques of the support set on all the datasets, separately along the inline and crossline directions of the test data, using support examples from the corresponding axis. From the results reported in Table 1, we observe that using more support examples is favorable for the F3 dataset along both directions. We do not observe much variations in performance between the sampling techniques of the support set for the Penobscot dataset. Thus, we use all the k examples in the support set for evaluation on the F3 and Penobscot datasets. However, the performance on the Parihaka dataset is more effective using the nearest slice in the support set along both directions due to the structural variations in the horizons. Therefore, we use the nearest slice to evaluate the Parihaka dataset for subsequent analyses.

Comparison with the baseline and other competing methods: We evaluate the performance of the AdaSemSeg and the baselines, Baseline-1 and Baseline-2, discussed in section 4.1 for all the datasets. From the results reported in Table 2, we observe that the Baseline-1, i.e., the AdaSemSeg trained only on the

Table 3. Comparison of the AdaSemSeg with ProtoSemSeg and transfer learning under the 1-shot and 5-shot set up. The best performance is highlighted in **bold**.

Target dataset	Metric	1−shot			5−shot		
		AdaSemSeg	ProtoSemSeg	Transfer learning	AdaSemSeg	ProtoSemSeg	Transfer learning
Parihaka inline	PA	**0.84**	0.51	0.49	**0.86**	0.56	0.59
	MCA	**0.75**	0.45	0.60	**0.76**	0.53	0.66
	FwIoU	**0.74**	0.36	0.37	**0.76**	0.43	0.45
	FwF_1	**0.84**	0.52	0.54	**0.86**	0.58	0.62
Parihaka crossline	PA	**0.82**	0.60	0.55	**0.84**	0.61	0.72
	MCA	**0.71**	0.42	0.66	**0.68**	0.58	0.74
	FwIoU	**0.71**	0.46	0.41	**0.74**	0.48	0.59
	FwF_1	**0.83**	0.62	0.58	**0.85**	0.64	0.74
Penobscot inline	PA	**0.92**	0.58	0.67	**0.95**	0.71	0.89
	MCA	**0.92**	0.45	0.75	**0.95**	0.62	0.93
	FwIoU	**0.85**	0.42	0.53	**0.91**	0.56	0.81
	FwF_1	**0.93**	0.58	0.67	**0.96**	0.71	0.89
Penobscot crossline	PA	**0.91**	0.54	0.62	**0.97**	0.70	0.87
	MCA	**0.90**	0.40	0.70	**0.95**	0.56	0.91
	FwIoU	**0.83**	0.38	0.47	**0.93**	0.56	0.76
	FwF_1	**0.90**	0.54	0.62	**0.96**	0.70	0.86
F3 inline	PA	**0.85**	0.57	0.83	**0.89**	0.69	0.85
	MCA	0.72	0.45	**0.80**	**0.79**	0.56	**0.79**
	FwIoU	**0.77**	0.40	0.73	**0.81**	0.53	0.75
	FwF_1	**0.85**	0.55	0.84	**0.89**	0.68	0.84
F3 crossline	PA	**0.87**	0.64	0.75	**0.87**	0.77	0.81
	MCA	**0.68**	0.36	0.62	**0.73**	0.44	0.70
	FwIoU	**0.79**	0.50	0.68	**0.80**	0.65	0.72
	FwF_1	**0.87**	0.64	0.79	**0.88**	0.77	0.82

target dataset, produces the best performance under most of the evaluation scenarios across all the datasets. For the F3 dataset along the crossline direction, the Baseline−2 outperforms all the methods under different evaluation metrics. Though the parameters of the AdaSemSeg are not tuned to interpret the images from the target dataset, it's performance is comparable to the baselines trained on the target datasets. We observe a marginal difference in the performance of the AdaSemSeg relative to the baselines on the Penobscot and F3 datasets. This suggests that the statistics in the support set are sufficient to make reasonable predictions on the query images. However, the difference with the baselines is noticeable in the Parihaka dataset, possibly due to its complexity. From the visual interpretation, the geological features in the Penobscot and F3 datasets appear simpler than the Parihaka dataset. Thus, the seismic features learned from the Penobscot and F3 datasets in the meta-training stage are not generalizing well to the Parihaka dataset. Using the complex Parihaka dataset in the meta-training stage is helping AdaSemSeg to produce impressive results on the F3 and Penobscot datasets. Overall, this experiment demonstrates the strength of AdaSemSeg in adapting to unseen classes in the target dataset. Fig. 4 illus-

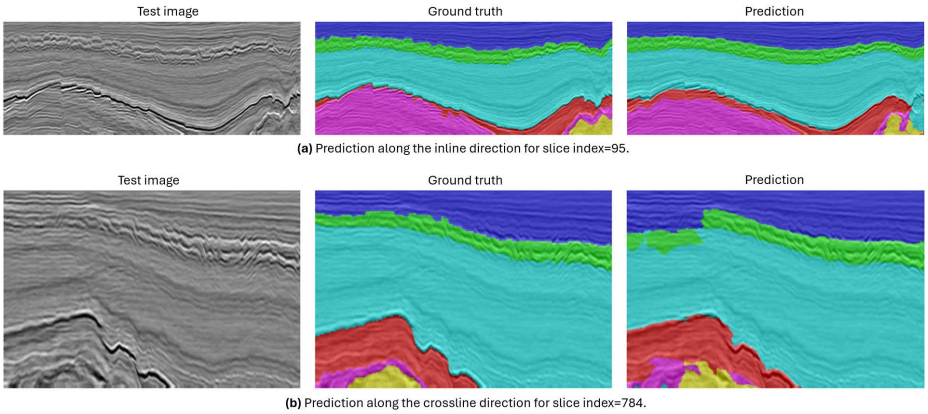


Fig. 4. Prediction of facies on the test data of the F3 dataset [1] along the inline and crossline directions by the AdaSemSeg trained in the 5−shot setup on the Parihaka and Penobscot datasets. The AdaSemSeg uses only 5− support examples (shown in the Appendix) to predict the facies on the unseen dataset.

trates the effectiveness of the AdaSemSeg in predicting facies on the unseen F3 dataset using the features learned from the Parihaka and Penobscot datasets. Predictions on other datasets and the corresponding support examples used for predicting facies are shown in the Appendix.

In another experiment, we compare the performance of the AdaSemSeg with ProtoSemSeg and transfer learning under 1−shot and 5−shot scenarios, outlined in Table 3. For the transfer learning, we use patches extracted from 1 and 5 slice(s) in the support set to train the segmentation network on the target dataset. However, the AdaSemSeg and ProtoSemSeg are *not* trained on the target dataset. The AdaSemSeg outperforms all the competing methods under multiple evaluation metrics. This experiment demonstrates the strength of the GP-based few-shot learning in the AdaSemSeg over the prototype-based few-shot segmentation method. Moreover, fine-tuning the segmentation network on the target dataset did not help in learning generalized representations that would assist in identifying facies in the target dataset.

Finetuning of the AdaSemSeg: We observed some scope for improvement for the AdaSemSeg on the Parihaka dataset. The seismic features learned from the Penobscot and F3 datasets are likely inconsistent with or are simple relative to the features observed in the Parihaka dataset. This presents a very realistic scenario where the target dataset can be more challenging relative to the source. In such occasions, *fine-tuning* the parameters to the target dataset is a plausible approach. We devised an experiment to find how many training slices are sufficient for the AdaSemSeg to achieve scores close to the baselines. The AdaSemSeg is fine-tuned in this experiment with $5, 10, 20$, and 50 training slices

Table 4. Evaluation of the AdaSemSeg on the Parihaka dataset after fine-tuning the AdaSemSeg on N slices from the training set of the Parihaka.

Number of training slices	inline				crossline			
	PA	MCA	FwIoU	FwF_1	PA	MCA	FwIoU	FwF_1
0	0.86	0.76	0.76	0.86	0.84	0.68	0.74	0.85
5	0.83	0.87	0.75	0.85	0.80	0.88	0.72	0.83
10	0.84	0.86	0.75	0.85	0.81	0.88	0.74	0.85
20	0.86	0.89	0.78	0.87	0.86	**0.91**	0.81	0.89
50	0.90	**0.91**	**0.83**	**0.90**	0.86	**0.91**	0.82	0.90
All	**0.91**	0.87	**0.83**	**0.90**	**0.91**	0.89	**0.85**	**0.92**

and evaluated under multiple metrics. As shown in Table 4, the metrics scores of the AdaSemSeg approach the baseline using 50 training slices.

Ablation study on the initialization of the image encoder: In another experiment, we study the effect of the image encoder's initialization on the AdaSemSeg's performance. In this experiment, we initialize the image encoder randomly and compare its performance with an image encoder trained using the SimCLR contrastive learning algorithm. We evaluate the performance of the proposed method on the Parihaka dataset under 1 and 5-shot scenarios, and the metric scores reported in Table 5 explain the importance of the initialization strategy on the performance of a few-shot algorithm. This observation is consistent with the performance of FSSS methods developed for natural images.

5 Conclusion

This paper presents a FSSS method using GP regressions that can adapt to different numbers of facies across datasets. We train the proposed method on three public datasets (having different numbers of facies) without making any

Table 5. Effect of the initialization strategy on the performance of the AdaSemSeg when evaluated on the Parihaka datasets using $k = \{1, 5\}$ support examples. The performance of the best method is presented in **bold**.

Shots	Initialization of the image encoder	inline				crossline			
		PA	MCA	FwIoU	FwF_1	PA	MCA	FwIoU	FwF_1
1	Random	0.61	0.58	0.48	0.64	0.56	0.50	0.42	0.59
	SimCLR	**0.84**	**0.75**	**0.74**	**0.84**	**0.82**	**0.71**	**0.71**	**0.83**
5	Random	0.72	0.66	0.59	0.72	0.55	0.57	0.43	0.60
	SimCLR	**0.86**	**0.76**	**0.76**	**0.86**	**0.84**	**0.68**	**0.74**	**0.85**

changes in the architecture for different datasets. The image encoder is initialized with the statistics learned from the seismic datasets studied in this work using a contrastive learning algorithm. This initialization strategy is more effective than random initialization and encourages the evaluation of the AdaSemSeg on unseen target datasets without the refinement of the parameters. The performance of the AdaSemSeg is comparable to the baselines trained on the target datasets, notably the F3 and Penobscot datasets. The AdaSemSeg outperforms another FSSS method and a segmentation network trained with transfer learning (fine-tuned on 1 and 5 slices from the target dataset). The comprehensive experimental evaluations on three datasets demonstrates the generalization capability of the AdaSemSeg to new seismic data with 1 or 5 annotated examples from the entire 3D volume and sets a new state-of-the-art for the FSSS of seismic facies.

References

1. Alaudah, Y., Michałowicz, P., Alfarraj, M., AlRegib, G.: A machine learning benchmark for facies classification. Interpretation **7**(3), 1A–T725 (2019)
2. Baroni, L., Silva, R.M., Ferreira, R.S., Civitarese, D., Szwarcman, D., Brazil, E.V.: Penobscot dataset: Fostering machine learning development for seismic interpretation (2021)
3. Chai, X., Nie, W., Lin, K., Tang, G., Yang, T., Yu, J., Cao, W.: An open-source package for deep-learning-based seismic facies classification: Benchmarking experiments on the seg 2020 open data. IEEE Transactions On Geoscience And Remote Sensing **60**(4507719), 1–19 (2022)
4. https://www.aicrowd.com/challenges/seismic-facies-identification challenge: Seg advanced modeling corporation ai project (March 17, 2024)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning (2020)
6. Chevitarese, D., Szwarcman, D., Brazil, E.V., Zadrozny, B.: Semantic segmentation of seismic images. In: International Joint Conference on Neural Networks (IJCNN) (2018)
7. Cunha, A., Pochet, A., Lopes, H., Gattass, M.: Seismic fault detection in real data using transfer learning from a convolutional neural network pre-trained with synthetic seismic data. Computers and Geosciences **135**, 104344 (2020)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
9. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning. vol. 70, pp. 1126–1135 (2017)
10. Gu, X., Cui, Y., Huang, J., Rashwan, A., Yang, X., Zhou, X., Ghiasi, G., Kuo, W., Chen, H., Chenz, L.C., Ross, D.: Dataseg: Taming a universal multi-dataset multi-task segmentation model. In: Advances in Neural Information Processing Systems (2023)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)

12. Johnander, J., Edstedt, J., Felsberg, M., Khan, F.S., Danelljan, M.: Dense gaussian processes for few-shot segmentation. In: European Conference on Computer Vision (2022)
13. Kaur, H., Pham, N., Fomel, S., Geng, Z., Decker, L., Gremillion, B., Jervis, M., Abma, R., Gao, S.: A deep learning framework for seismic facies classification. Interpretation **11**(1), 1F–Z4 (2023)
14. Lambert, J., Liu, Z., Sener, O., Hays, J., Koltun, V.: Mseg: A composite dataset for multi-domain semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
15. Monteiro, B.A.A., Oliveira, H., Santos, J.A.d.: Self-supervised learning for seismic image segmentation from few-labeled samples. IEEE Geoscience and Remote Sensing Letters **19**, 1–5 (2022)
16. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. The MIT Press (2006)
17. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: International Conference on Learning Representations (2017)
18. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention. pp. 234–241 (2015)
19. Saha, S., Choi, O., Whitaker, R.: Few-shot segmentation of microscopy images using gaussian process. In: MOVI, MICCAI (2022)
20. Saha, S., Gazi, W., Mohammed, R., Rapstine, T., Powers, H., Whitaker, R.: Multitask training as regularization strategy for seismic image segmentation. IEEE Geoscience and Remote Sensing Letters **20**, 1–5 (2023)
21. Salles Civitarese, D., Szwarcman, D., Silva, R., Vital Brazil, E.: Transfer learning applied to seismic images classification. In: AAPG ACE (05 2018)
22. Shaban, A., Shray, Liu, B.Z., Essa, I., Boots, B.: One-shot learning for semantic segmentation. In: British Machine Vision Conference. BMVA Press (2017)
23. Shang, C., Li, H., Meng, F., Wu, Q., Qiu, H., Wang, L.: Incrementer: Transformer for class-incremental semantic segmentation with knowledge distillation focusing on old class. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7214–7224 (2023)
24. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems. vol. 30 (2017)
25. Tian, Z., Zhao, H., Shu, M., Yang, Z., Li, R., Jia, J.: Prior guided feature enrichment network for few-shot segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(2), 1050–1065 (2022)
26. Tolstaya, E., Egorov, A.: Deep learning for automated seismic facies classification. Interpretation **10**(2), B13–T392 (2022)
27. Wang, Z., Jiang, Z., Yuan, Y.: Prototype queue learning for multi-class few-shot semantic segmentation. In: IEEE International Conference on Image Processing (ICIP). pp. 1721–1725 (2022)
28. Zhao, Y., Chai, B., Shuo, L., Li, Z., Wu, H., Wang, T.: Few-shot learning for seismic facies segmentation via prototype learning. GEOPHYSICS **88**(3), 41–49 (2023)
29. Zhou, Q., Liu, Y., Yu, C., Li, J., Wang, Z., Wang, F.: Lmseg: Language-guided multi-dataset segmentation. In: International Conference on Learning Representations (2023)

Appendix to AdaSemSeg: An Adaptive Few-shot Semantic Segmentation of Seismic Facies

Surojit Saha[1] and Ross Whitaker[1]

Scientific Computing and Imaging Institute, Kahlert School of Computing, The University of Utah, Salt Lake City, USA {surojit.saha,ross.whitaker}@utah.edu

1 Details of the Neural Network Architectures

Fig. 1 shows the detail of the convolutional filters, batch normalization layers, and activation functions used in the image encoder (IE), mask encoder (ME), and decoder (D) of the AdaSemSeg. The IE is the ResNet50 [5], and the decoder is similar to that used in the U-Net [7] with double convolution on the concatenated data followed by transpose convolution. The GP regression takes as input the deep-encoded image features and encoded mask features having the same spatial resolutions. The GP regression is used in two latent layers, i.e., at the bottleneck and the layer above it, whose predictions are fed to the decoder. In addition, the decoder uses shallow encoded image features to predict the binary mask for a query image using support examples.

The Baseline-2 and ProtoSemSeg use the IE and D used in the AdaSemSeg with skip connections at multiple layers of encoding as shown in Fig. 2. Essentially, this is a U-Net with the ResNet as the encoder, and we call this the ResNet-UNet model. In the case of the ProtoSemSeg [8], the ResNet-UNet predicts binary mask $C = 2$, similar to the AdaSemSeg, and the total loss is accumulated across all the classes in a dataset. However, depending on the dataset, the ResNet-UNet is fixed to $C = 6, 7$ classes for the Baseline-2.

2 Experiments

2.1 Training the SimCLR

The image encoder is trained using SimCLR[4] on a total of 35648 patches extracted from the three datasets. Following augmentations used for producing the positive and negative samples: rotation ($[-20°, 20°]$), horizontal flip, Gaussian blur ($[0.1, 2.0]$), Gaussian noise ($[1e-4, 5e-2]$), random crop with resize, brightness ($[0.5, 1.5]$), and contrast ($[0.0, 2.0]$). The temperature parameter of the SimCLR is set as $\tau = 0.07$. The batch size used in this work is 32, and we did not observe any performance benefits from a bigger batch size. We used the Adam optimizer with a learning rate of $lr = 3e-04$ and weight decay of $1e-04$, and the model was trained for 10 epochs. The SimCLR achieves an accuracy of 93.75% and 98.44% under top-1% and top-5% evaluation criteria, respectively.

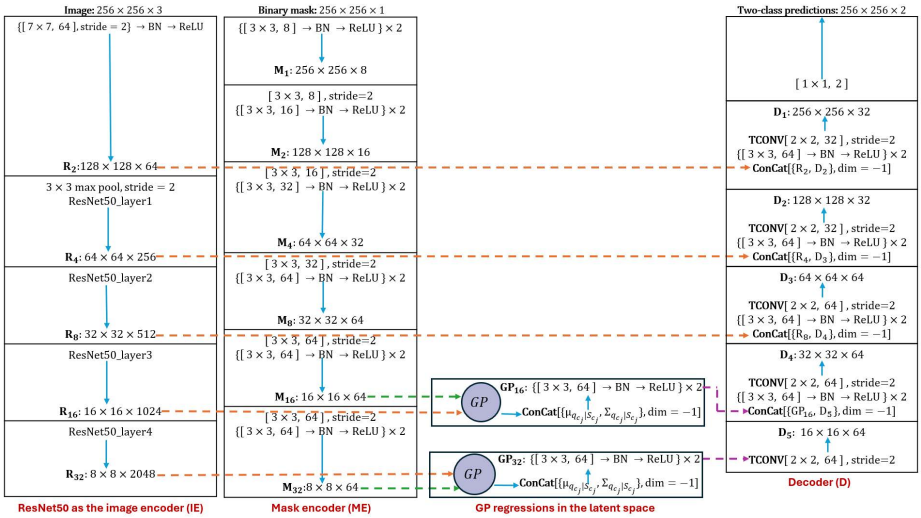


Fig. 1. The neural network architecture of the AdaSemSeg used to predict the binary mask for a query image using support examples.

2.2 Training the AdaSemSeg

The GP regression used in the AdaSemSeg uses the configuration used in the DGPNet [6]. The AdaSemSeg is trained on patches extracted from three datasets. We use 8100, 7,664, and 8,206 patches extracted from the Penobscot, F3, and Parihaka datasets to train the AdaSemSeg under different settings discussed in the paper. We follow the leave-one-out policy to train the AdaSemSeg in a few-shot setup. For example, the source dataset used in the meta-training is built using the Penobscot and F3 datasets, and the trained model is evaluated on the unseen Parihaka dataset. The data augmentations used in this experiment are the RandomRotate ($[-20°, 20°]$), RandomHorizontalFlip, GaussianBlur ($[0.1, 2.0]$), and GaussNoise ($[1e-4, 5e-2]$). The image encoder is initialized with the SimCLR-trained statistics. The AdamW optimizer with the learning rate, $lr = 5e-05$, and weight decay of $1e-03$ is used in this experiment. The learning rate is reduced to 0.25 of the existing rate when the validation loss does not improve for 5 epochs.

2.3 Training the Baselines and Competing methods

The baseline methods used in this work, Baseline-1 and Baseline-2, and the competing methods, ProtoSemSeg [8] and transfer learning, use the same amount of training data as the AdaSemSeg, under different evaluation scenarios. For the different comparisons studied in this work, all the methods are trained for the same number of epochs with the same configuration for the optimizer and learning rate scheduler. For the ProtoSemSeg, we refer to the settings reported in [8] for all the experimental evaluations.

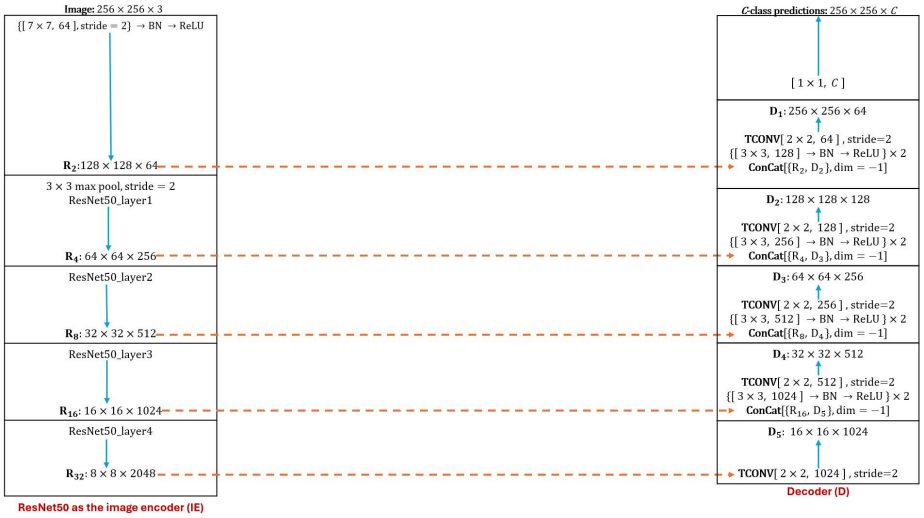
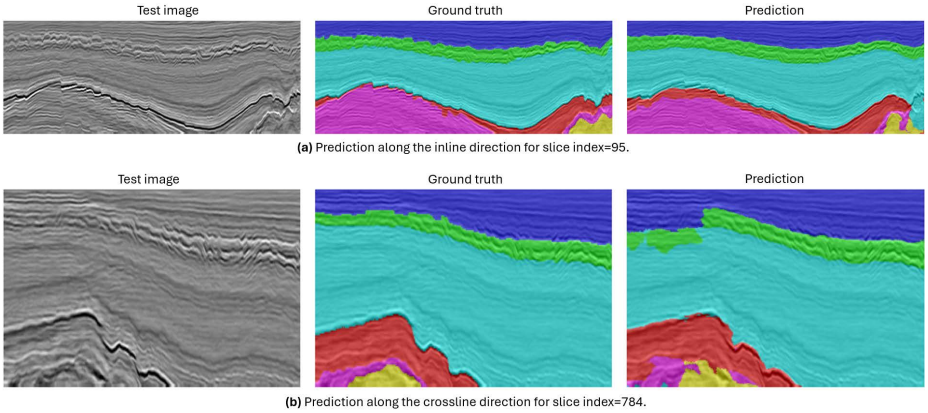


Fig. 2. The neural network architecture used in Baseline-2 and ProtoSemSeg used to predict the multi-class (C) mask.

3 Additional Results

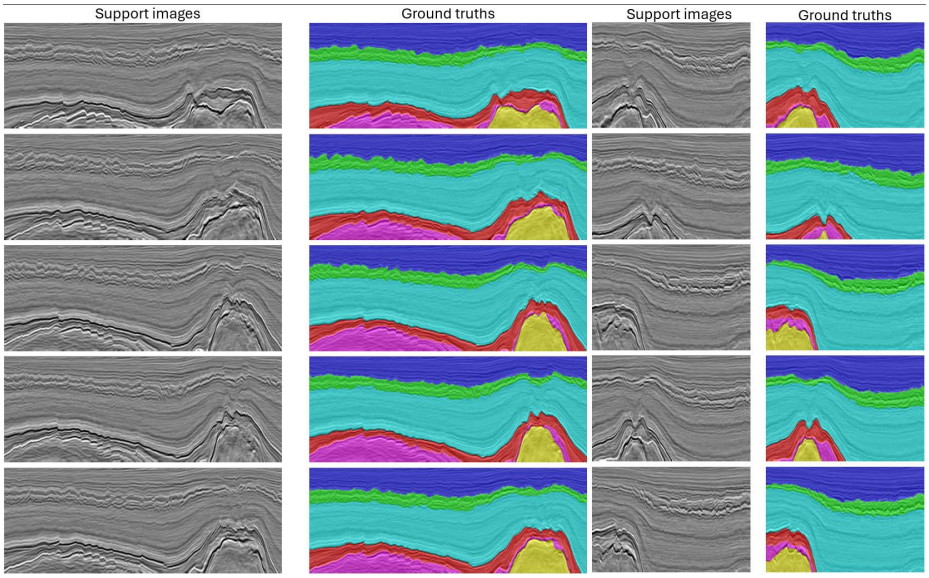
In this section, we discuss the predictions on the F3, Parihaka, and Penobscot datasets by the AdaSemSeg. Fig. 3 illustrates the predictions on test slices along the inline and crossline directions using the corresponding support set shown in Fig. 4. The predicted facies are very close to the ground truth, except for the complex structural variations. For the F3 dataset, we use all the support examples for the predictions due to the structural similarity across slices in both directions. However, we observe a lot of variation across slices for the Parihaka dataset along both axes, as shown in Fig. 5. Thus, we resort to the idea of *nearest slice* for the evaluation of test slices, as shown in Fig. 7. Thus, the support set has a single example along each direction, slice index=$\{525\}$ along the inline axis and slice index=$\{657\}$ along the crossline axis. The predictions on the Parihaka dataset are shown in Fig. 6. The AdaSemSeg does a reasonable job of recognizing the complex facies structure *without* training the network parameters on the target dataset. The prediction on the Penobscot dataset is shown in Fig. 8.



(a) Prediction along the inline direction for slice index=95.

(b) Prediction along the crossline direction for slice index=784.

Fig. 3. Prediction of facies on the test set of the F3 dataset [1] along the inline and crossline directions by the AdaSemSeg trained in the 5−shot setup on the Parihaka and Penobscot datasets. The AdaSemSeg uses only 5− support examples (shown in Fig. 4) to predict the facies on the unseen dataset.



(a) Support set along the inline direction.

(b) Support set along the crossline direction.

Fig. 4. The support set of the F3 facies data [1] that spans through the entire volume both along the inline (slice indices=$\{15, 111, 207, 303, 400\}$) and crossline (slice indices=$\{0, 171, 343, 514, 686\}$) directions. The AdaSemSeg uses the support examples to predict facies on the test slices of the F3 as shown in Fig. 3. We observe structural similarity across slices in both directions.

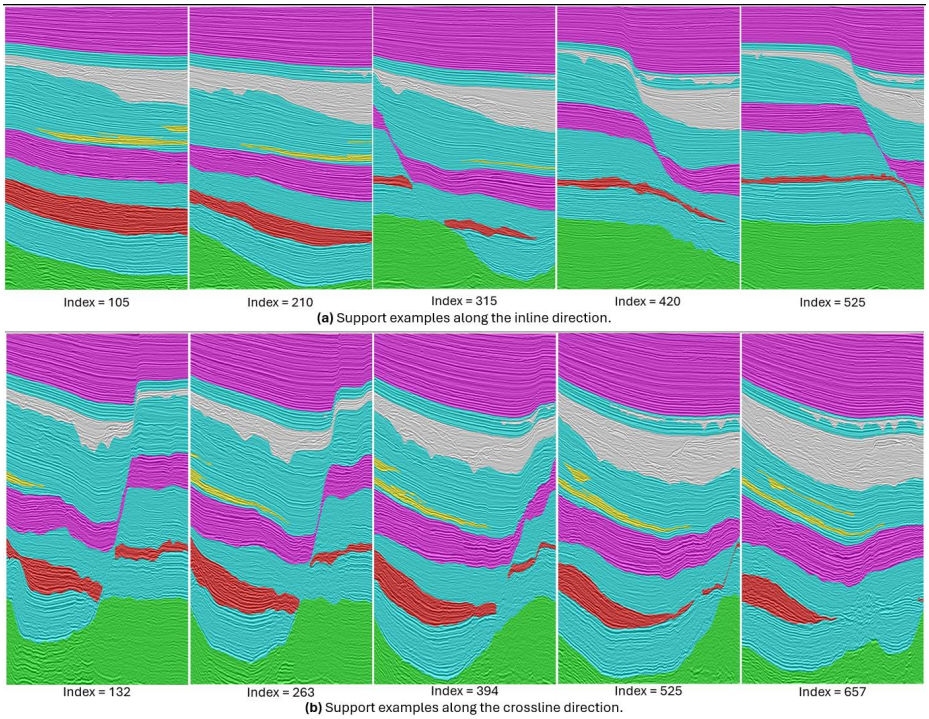


Fig. 5. The support set of the Parihaka dataset [3] that spans through the entire volume both along the inline (slice indices=$\{[105, 210, 315, 420, 525]\}$) and crossline (slice indices=$\{132, 263, 394, 525, 657\}$) directions. For clarity, we overlay the input with the ground truth annotations. We observe a lot of structural variability across slices in both directions. Thus, we use a slice closest to the query image to predict facies.

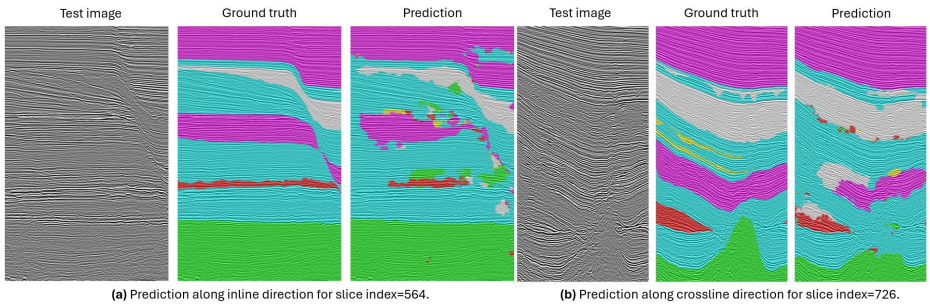


Fig. 6. Prediction of facies on the test set of the Parihaka dataset [3] along the inline and crossline directions by the AdaSemSeg trained in the 5−shot setup on the F3 and Penobscot datasets. The AdaSemSeg uses only 1− support examples (as shown in Fig. 7 to predict the facies on the unseen dataset.

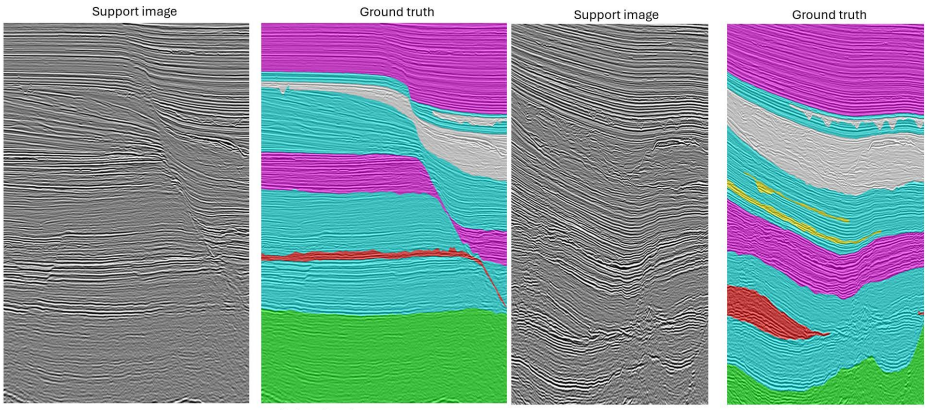


Fig. 7. The support set for the Parihaka dataset [3] is the nearest slice to the test data both along the inline (slice index=$\{525\}$) and crossline (slice index=$\{657\}$) directions due to the structural variations along both axes (refer to Fig. 5). The AdaSemSeg uses the support example along each axis to predict facies, as shown in Fig. 6.

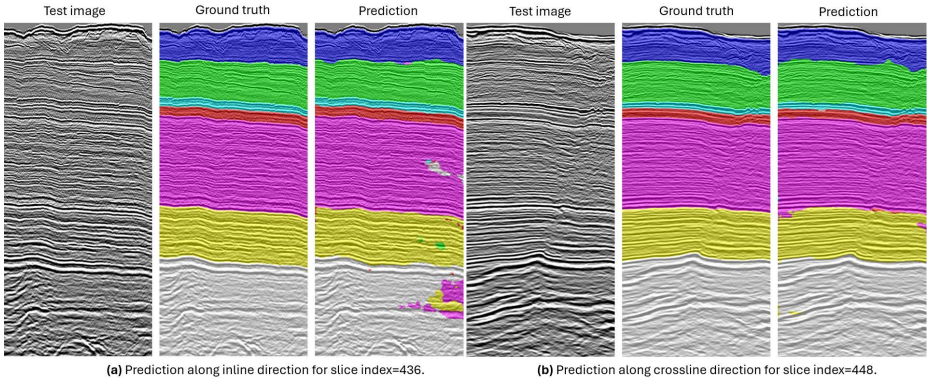


Fig. 8. Prediction of facies on the test data of the Penobscot dataset [2] along the inline and crossline axes by the AdaSemSeg trained in the 5−shot setup on the F3 and Parihaka datasets. The AdaSemSeg uses 5− support examples, similar to the F3 dataset, to predict the facies on the unseen dataset.

References

1. Alaudah, Y., Michałowicz, P., Alfarraj, M., AlRegib, G.: A machine learning benchmark for facies classification. Interpretation **7**(3), 1A–T725 (2019)
2. Baroni, L., Silva, R.M., Ferreira, R.S., Civitarese, D., Szwarcman, D., Brazil, E.V.: Penobscot dataset: Fostering machine learning development for seismic interpretation (2021)
3. https://www.aicrowd.com/challenges/seismic-facies-identification challenge: Seg advanced modeling corporation ai project (March 17, 2024)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning (2020)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
6. Johnander, J., Edstedt, J., Felsberg, M., Khan, F.S., Danelljan, M.: Dense gaussian processes for few-shot segmentation. In: European Conference on Computer Vision (2022)
7. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention. pp. 234–241 (2015)
8. Zhao, Y., Chai, B., Shuo, L., Li, Z., Wu, H., Wang, T.: Few-shot learning for seismic facies segmentation via prototype learning. GEOPHYSICS **88**(3), 41–49 (2023)