Make me an Expert: Distilling from Generalist Black-Box Models into Specialized Models for Semantic Segmentation

Yasser Benigmim[1,2] Subhankar Roy[3] Khalid Oublal[1] Imad Eddine Marouf[1] Slim Essid[4*]
Vicky Kalogeiton[2] Stéphane Lathuilière[5]
[1] LTCI, Télécom-Paris, Institut Polytechnique de Paris, France
[2] LIX, Ecole Polytechnique, CNRS, Institut Polytechnique de Paris
[3] University of Bergamo, Italy [4] NVIDIA [5] Inria at University Grenoble Alpes, LJK, France

Abstract

The rise of Artificial Intelligence as a Service (AIaaS) democratizes access to pre-trained models via Application Programming Interfaces (APIs), but also raises a fundamental question: how can local models be effectively trained using black-box models that do not expose their weights, training data, or logits, a constraint in which current domain adaptation paradigms are impractical ? To address this challenge, we introduce the Black-Box Distillation (B^2D) setting, which enables local model adaptation under realistic constraints: (1) the API model is open-vocabulary and trained on large-scale general-purpose data, and (2) access is limited to one-hot predictions only. We identify that open-vocabulary models exhibit significant sensitivity to input resolution, with different object classes being segmented optimally at different scales, a limitation termed the "curse of resolution". Our method, ATtention-Guided sCaler (ATGC), addresses this challenge by leveraging DINOv2 attention maps to dynamically select optimal scales for black-box model inference. ATGC scores the attention maps with entropy to identify informative scales for pseudo-labelling, enabling effective distillation. Experiments demonstrate substantial improvements under black-box supervision across multiple datasets while requiring only one-hot API predictions. Our code is available at https://github.com/yasserben/ATGC.

1. Introduction

The paradigm of pre-training neural networks on large datasets [4, 13, 57] has produced powerful "Foundation Models" (FMs) [6] with broad applicability across numerous domains [7, 19, 35, 40, 58, 68]. Many of these FMs are commercialized as Artificial Intelligence as a Service (AIaaS) and accessed through APIs, such as GPT-4 [1] and Gemini [15]. While AIaaS simplifies infrastructure management, it presents users with significant challenges, including substantial costs at scale [11], network latency, and potential downtime.

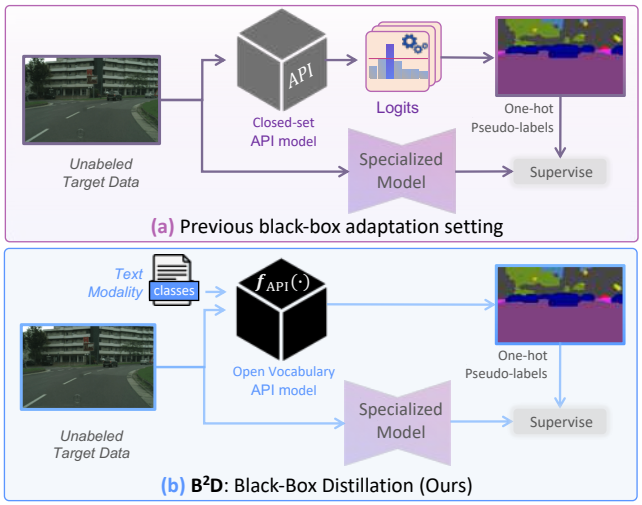


Figure 1. **Comparison of black-box adaptation settings**. **(a)** Previous approaches assume access to API logits when leveraging pseudo-labels for student model training, which makes them "gray-box". **(b)** Our proposed **B**lack-**B**ox **D**istillation (B^2D) setting defines a more realistic "black-box" scenario, using open-vocabulary APIs without any access to logits.

A viable alternative is knowledge distillation (KD) [22, 56], where knowledge from a generalist *teacher* is distilled into a compact *student* model. This creates a local "expert"[1] model that is cost-effective by eliminating calls to external APIs and can be finetuned for downstream tasks. Such models can be created through standard KD if the teacher is *open-weight* [26], or via black-box adaptation for *closed-*

*This work was conducted while the author was at Télécom Paris.

[1] Our usage of the term "expert" differs from the KD literature, where expert model refers to the teacher, but closer to mixture of experts [31].

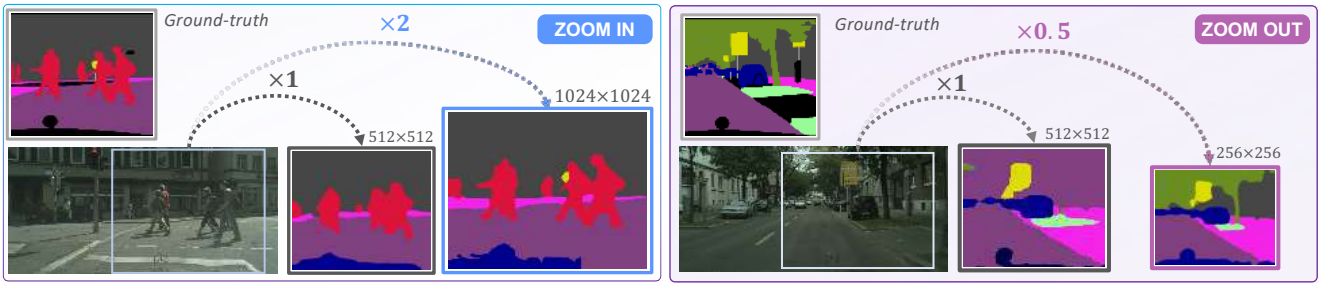


Figure 2. **Scale-dependent segmentation quality**. We observe that segmentation performance varies with input resolution. Zooming in ($\times 2$, **left**) enhances segmentation of small, distant objects like pedestrians, while zooming out ($\times 0.5$, **right**) improves large contextual elements by capturing broader spatial relationships. We refer to this issue as "*curse of resolution*" where no single resolution optimally segments all object classes. This highlights the challenges inherent to distilling knowledge from open-vocabulary generalists models.

weight teachers accessible only through an API [18]. We focus on the latter as it is both more challenging since it excludes techniques like weight initialization or intermediate layer distillation [32, 65], and more realistic given the growing prevalence of commercial API-based foundation models.

Existing black-box adaptation methods often leverage pseudo-labels (PLs) from an API model while assuming access to its output logits [18, 70]. While these approaches are termed "black-box", we believe that "gray-box" (see Fig. 1(a)) would be a more appropriate description, given the access to logits. The "gray-box" assumption is a notable limitation, as commercial APIs increasingly withhold logits to protect intellectual property. This makes truly black-box approaches, which have seen recent interest for Large Language Models (LLMs) [24, 39, 45, 55], more relevant.

With a similar goal in mind, in this work, we formalize and study the problem of distilling a generalist black-box API-based foundation model into an expert model, but for the task of semantic segmentation. In this new setting, what we call **B**lack-**B**ox **D**istillation (B^2D), we assume that the API model is truly black-box and serves an open-vocabulary semantic segmentation model, which we can prompt using natural language [33, 63] to extract pixel-level semantic labels. As shown in Fig. 1(b), we use these PLs from the teacher to train the student model. The newly introduced B^2D is more challenging because: (i) strict black-box assumption limits the scope of PL filtering, and (ii) the resolution of the image crop presented to the API governs the quality of PLs (see Fig. 2). Since the pre-training data and their statistics are never disclosed for strictly-private API models, querying the API with optimal resolution to obtain high-quality PL is non-trivial.

To tackle B^2D, we propose a student-teacher self-training framework designed to obtain higher quality PLs from the API model. In the absence of confidence thresholding, we employ a simple strategy of *prompt ensembling*, where we prompt the API using an averaged embedding derived from various synonyms of a noun (*e.g.*, "train", "tram", "locomotive"). To determine the optimal image crop resolution for the API, we introduce a novel module, **AT**tention-**G**uided s**C**aler (ATGC), that dynamically identifies the most suitable crop resolution. Our approach leverages the common finding in semantic segmentation literature that different object classes are best segmented at different image resolutions [12, 29]. For instance, "distant traffic lights" are better segmented with *zoomed-in* crops, while "closer trucks" benefit from the larger context provided by *zoomed-out* crops. We objectively assess the *goodness* of a crop resolution using the attention maps of the student model, which we score using Shannon entropy. A lower entropy value indicates a more focused map, suggesting that the resolution is well-suited to the objects within the crop. To ensure meaningful attention maps, we initialize the student's encoder with an open-weight model (*e.g.*, DINOv2 [46]) and keep it frozen during training. ATGC then selects the crop yielding the lowest entropy score and feeds it to the API model to obtain the PLs. Our contributions can be summarized as follows:

- We formalize **B**lack-**B**ox **D**istillation (B^2D), a novel setting which assumes no access to API logits in contrast to previous "gray-box" approaches, better reflecting the reality of commercial AI services.
- We propose **AT**tention-**G**uided s**C**aler (ATGC), a novel model which uses DINOv2 attention maps to dynamically select optimal image resolutions for API queries, yielding higher-quality pseudo-labels.

We benchmark our framework on Cityscapes [16] and ACDC [53], demonstrating superior performance over state-of-the-art methods. Our findings highlight both the challenges and potential of specializing models from truly black-box APIs.

2. Related work

We review the literature related to B^2D, but limit our discussion to works that tackle the downstream task of semantic segmentation, which is the scope of this work.

Black-box adaptation refers to adapting from a pretrained model to a new domain or task without accessing or modifying its internal parameters, treating the model as a "black-box". In semantic segmentation, the precursors to black-box adaptation include unsupervised domain adaptation (UDA), where both the source dataset and source model are available during adaptation [9, 25–27, 47], and source-free UDA (SFUDA) [20, 36, 43], which assumes access to only the source model, but not the source data. Both of these settings can be considered as "white-box", as they allow access to the weights of the source model. Although white-box adaptation offers maximum flexibility and control, it also introduces several challenges such as security risks [38, 59] or weights being unavailable due to commercialization [1].

In contrast, black-box UDA (B^2UDA) discards the assumption of access to the source pretrained model's weights, and instead treats it as a black box, accessible only via an API. B^2UDA has mainly been studied in the context of image classification [38, 70, 71], and has only recently been investigated for semantic segmentation [18]. In detail, Cuttano *et al*. [18] proposed a mechanism to extract reliable PLs from the black-box model through confidence-based filtering. A downside of this approach is that it involves accessing the logits from the API, which may not always be guaranteed, and thus can more accurately be considered as a "gray-box" approach [55]. Differently, in our proposed B^2D setting, we adhere to the true definition of a black-box setting – neither the source model's weights nor the output logits are available. The key differences among the various settings have been summarized in Tab. 1.

Distilling from foundation models. Foundation models (FMs) are large-scale general-purpose models [3, 35, 46, 48] trained on massive, diverse datasets, designed to serve as a solid "foundation" or starting point for various downstream tasks [3]. While the FMs offer many opportunities [6], their large memory footprint hinder deployment on resource-constrained devices [50]. To balance performance and efficiency, Knowledge Distillation (KD) [23] has been adopted as a go-to technique to train a smaller student model using a FM as a teacher. At its core, KD-based approaches assume that the teacher is open-weight (*i.e*., white-box), and thus distill knowledge either using the teacher's output logits [5, 44, 46], internal feature representations [60, 66, 67, 69] or their combination [42, 56]. Very recently, this idea has been extended to distill knowledge from multiple FMs into a single student model [21, 49, 54], and has demonstrated strong performance in multiple tasks, including semantic segmentation. Despite the progress enabled by KD, the white-box (or gray-box) assumption makes it inapplicable in truly restrictive APIs. The B^2D setting proposed in this work is challenging as it allows only a "hard" variant of KD, through the use of one-hot PLs.

Unlike B^2UDA [18], which allows adaptation to a fixed set of categories (or vocabulary), distilling from open-vocabulary FMs is more appealing, as joint training with vision and language modalities allow for adaptation beyond a fixed vocabulary through flexible natural language prompts (*e.g.*, `a photo of a [CLASS]` [33, 48]), facilitating deployment of the student model to any domain/task of interest. However, this advantage may get eclipsed in certain scenarios when the target domain (*e.g.*, medical images) exhibits a significant domain gap with respect to the FM pre-training dataset. This will lead to very noisy PLs [10], especially since the pre-training dataset is often not disclosed by API providers. To balance flexibility and performance, we propose two strategies to extract more reliable PLs, while staying within the scope of the true black-box setting.

Settings	Source data	Pretrained model	Deployment feasibility
UDA [26]	✓		Low
SFUDA [43]	×		Moderate
B^2UDA [18]	×		Moderate
B^2D (Ours)	×		High

Table 1. **Different model adaptation settings.** (, ,) denote white-box, gray-box and black-box models, respectively. B^2UDA methods assume access to logits, whereas our B^2D does not. (,) denote closed-set and open-vocabulary pre-trained source model. Due to the use of open-vocabulary models, B^2D allows adaptation to any desired set of classes, making it highly flexible for deployment on downstream tasks.

3. Problem formulation and preliminaries

The goal of Black-Box Distillation (B^2D) is to transfer knowledge from a *generalist black-box* model, accessed via an API, to a *local* model. Next, we formulate the problem and introduce some preliminaries.

Problem setup. We are interested in training a local model for the task of semantic segmentation. We assume that we have access to an API that serves an open-vocabulary segmentation model $\boldsymbol{f}_{\text{API}}$, which can produce *one-hot* semantic segmentation map when presented with an image $X \in \mathbb{R}^{3\times h\times w}$. We denote the one-hot segmentation map as $\mathcal{M} = \{0,1\}^{|\mathcal{C}|\times h\times w}$, where $\mathcal{C} = \{c_1, c_2, \cdots, c_K\}$ is the set of K class names (or vocabulary). Note, $\mathcal{C}$ is not a set of class indices, but strings (*e.g.*, "*car*", "*pedestrian*"), and is provided by the user.

Given a dataset of unlabelled images from the target domain, $\mathcal{D} = \{X_i\}_{i=1}^n$, our goal is to learn a local segmentation model $\mathcal{F}_\theta \colon X \mapsto [0,1]^{|\mathcal{C}|\times h\times w}$ that infers per-pixel class probabilities. The only supervision available for learning the parameters θ of $\mathcal{F}_\theta$, is the API model $\boldsymbol{f}_{\text{API}}$.

Challenges in B^{2}D. While black-box adaptation is challenging in itself, B^{2}D is more challenging than B^{2}UDA [18] in two key aspects: (**i**) B^{2}UDA assumes that the API model was trained on a niche labelled source dataset (*e.g.*, GTA [51]) that has a relatively small domain gap with the target domain (*e.g.*, Cityscapes). We argue that this strong assumption does not reflect real-world domain gaps, and as a result, the adaptation strategy may not generalize beyond academic benchmarks. Instead, in B^{2}D, we propose to leverage an open-vocabulary segmentation model as the API. This choice can be viewed as a "double-edged sword", since, unlike B^{2}UDA, it offers the flexibility to distill information for any vocabulary, but at the same time may widen the domain gap when the target domain is very different from the (unknown) pre-training data distribution. (**ii**) Our setting operates under the stricter assumption that the API provides only one-hot segmentation maps ($\mathcal{M}$). This constraint is motivated by the practical limitations of many real-world APIs, which often return only final predictions. It was shown in [23] that KD derives its strength from the use of "soft targets" alongside hard labels, preventing overfitting. Thus, the absence of privileged information in B^{2}D prevents the use of effective techniques, such as confidence-based pseudo-labelling [18] or soft-dillation [56], to mitigate the impact of noisy PLs, making it a more challenging task.

Preliminaries. Knowledge distillation [23] consists in training a smaller (student) model to mimic the behaviour of a larger (teacher) model. The idea is to compress the knowledge of the teacher into a lightweight, faster model, without losing much performance. This is achieved through a distillation loss $\mathcal{L}_{\text{KD}}$ that is a sum of KL-divergence loss, computed between the teacher's p_t, and the student's p_s predicted probability distributions, and a cross-entropy loss between p_s and the true hard label Y of the sample:

$$\mathcal{L}_{\text{KD}} = \alpha \text{KL}(p_t \| p_s) + (1-\alpha)\mathcal{L}_{\text{CE}}(p_s, Y), \qquad (1)$$

where α is a hyperparameter. In B^{2}D, we do not have access to p_t, preventing us from directly using the formulation in Eq. (1) to train $\mathcal{F}_\theta$.

4. Methods

Motivation. In this work, we argue that the quality of the segmentation maps $\mathcal{M}$ derived from the API model $\boldsymbol{f}_{\text{API}}$ is influenced by the resolution of the input image. To verify this, we conducted a preliminary study in which we varied the resolution (or scale[2]) of Cityscapes images and fed them to an API model that serves an open-vocabulary segmentation model (SAN [64]). We prompted the API using the class names of Cityscapes and report in Fig. 3 the class-wise intersection over union (IoU) performance for a selected few classes. From Fig. 3 we observe that IoU for a class varies across scales, and there is no single scale that produces the best IoU across classes. For example, classes such as "road" and "bus" that cover a significant area of the scene require larger contextual information and are hence better segmented at zoomed-out resolution (or scale factors < 1). Conversely, some classes that are typically distant and cover a tiny area of the scene (*e.g.*, "traffic sign" and "pole") show peak performance when zoomed in (or higher resolution, with scale factors > 1). This happens because small or distant objects do not require very long-range information to be well segmented. More qualitative and quantitative examples are provided in Appendices C.2 and C.4

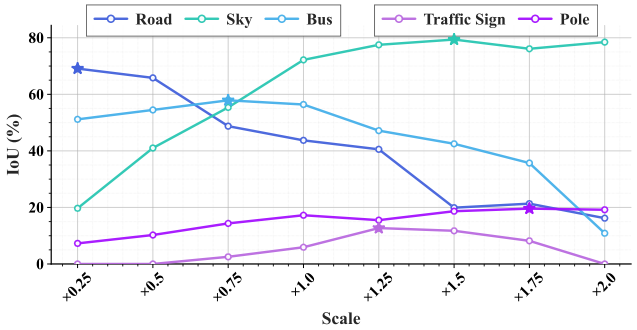


Figure 3. **Impact of scale on class-wise IoU performance.** The plot shows that performance varies across scales and across classes: larger-scale objects like "road" have peak performance at a lower resolution ($\times 0.5$), while smaller-scale, distant objects like "traffic sign" is better segmented at higher resolutions ($\times 1.75$).

This phenomenon called *the curse of resolution*, also noted in [29, 72], is exacerbated when using an open-vocabulary segmentation model as the API, since the predominant resolution of its training images are likely to be different from the target dataset. The high variability in performance (*e.g.*, IoU for the class "road" drops from $\sim 70\%$ to $\sim 20\%$ when scale is varied) clearly indicates that identifying an optimal resolution can significantly improve the quality of PLs, ultimately leading to a high-performing local segmentation model. This insight forms the foundation of our proposed method. Next, we give a brief overview of our method and then describe the novel component in detail.

Method overview. Our goal is to train a local segmentation model that becomes an "expert" at segmenting target images by using supervision from a "generalist" black-box API model, which in our case is an open-vocabulary segmentation model. We adopt a student-teacher framework [18, 28], where the API model acts as a teacher and the local segmentation model acts as a student. The teacher supervises the student model by providing one-hot PLs to train its parameters. However, differently from [18] we do

[2]An image is scaled by an arbitrary factor, and crops of the size expected by the network are extracted using a sliding window. We use scale and resolution interchangeably.

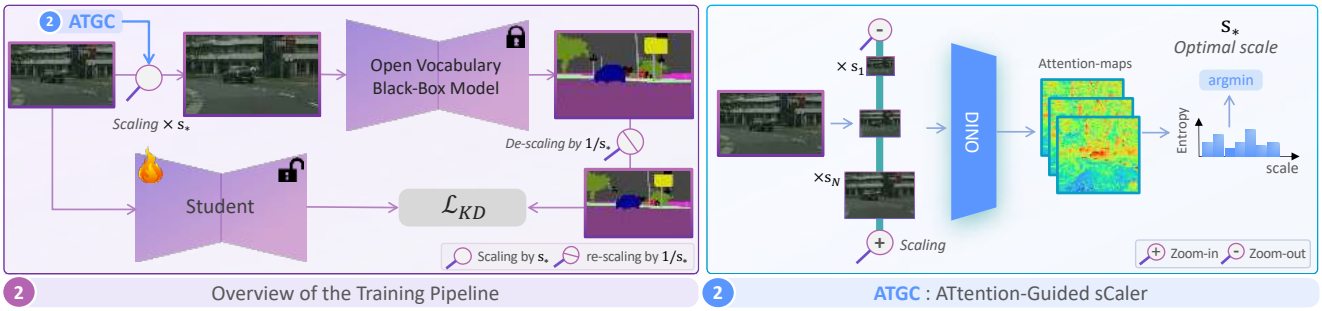


Figure 4. **Overview of ATGC and our training pipeline.** (**Left**) (1) Training pipeline overview: Our approach operates under minimal access, relying solely on one-hot predictions $(0/1)$ from the open-vocabulary black-box model. The student network is trained using knowledge distillation loss L_{KD} with pseudo-labels generated at optimal scales determined by our ATGC. (**Right**) (2) ATGC (**AT**tention-**G**uided s**C**aler) detailed process: For each input image, multiple crops are generated and scaled to different resolutions. DINOv2 attention maps are computed for each scale, and entropy-based scoring identifies the optimal scale S^* that yields the most informative features. The selected scale is used to query the black-box model, generating high-quality one-hot pseudo-labels that are re-scaled and used as supervision targets for student model training.

not employ an exponential moving average (EMA) teacher derived from the student. Instead, we simply focus on extracting higher quality PLs to supervise the student.

As shown in Fig. 4, at the core of our student-teacher framework is a novel module, ATGC (**AT**tention-**G**uided s**C**aler). We designed ATGC to identify (or *mine*) the optimal scale that produces less noisy PLs than those obtained through naive approaches such as random scaling. ATGC exploits the attention maps of the student as a proxy to determine whether a given scale is suitable for segmenting the objects contained in a crop. We find that well-defined and crisp attention maps strongly correlate with superior dense prediction performance. Through a scoring mechanism, we pick the scale corresponding to the best attention map, and use that scale to preprocess the image crop and query the API to get PLs for supervision.

4.1. ATtention-Guided sCaler (ATGC)

The proposed ATGC is a *plug-and-play* module, applied on a given image crop before being fed to the API, whose goal is to choose the optimal scale. While a naive approach would be to perform random scaling (as in random resized crop data augmentation), we argue that it has two downsides: ***(i)*** as discussed before, not all objects are well-segmented at every scale, and this will contribute to noisier PLs, which will be detrimental for the student when trained for a longer period [41]; and ***(ii)*** it incurs unnecessary API calls, increasing cost due to inefficient queries. Therefore, ATGC facilitates learning in terms of both stable training and budget.

We design ATGC drawing inspirations from self-supervised learning methods [2, 46], which have shown that Vision Transformers (ViTs) trained with self-supervision can yield strong localization properties. To exploit this property of ViTs we employ a pre-trained DINOv2 [46] as the student encoder. In particular, we leverage the attention maps between the [CLS] token and the patch tokens from the final layer of DINOv2, and consider them a *proxy indicator* of the suitability of a given image crop for pseudo-labelling. Next, we describe how we extract the attention maps, score them, and use the scores to find the optimal scale. We subsequently discuss why our design philosophy of utilizing attention maps works.

Extracting Attention Maps. Given an image crop $\mathbf{X} \sim \mathcal{D}$, we scale it to $\mathbb{N}$ different resolutions using a set of scale factors defining the support $\mathcal{S} = \{\mathrm{s}_j \mid j \in \mathbb{N}, \mathrm{s}_{min} \leq \mathrm{s}_j \leq \mathrm{s}_{max}\}$, where $\mathrm{s}_{max}, \mathrm{s}_{min} \in \mathbb{R}^+$ are the maximum and minimum values. We feed each scaled image $\mathbf{X}_j = \mathcal{T}_{\mathrm{s}_j}(\mathbf{X})$ (where $\mathcal{T}_{\mathrm{s}_j} : \mathbf{X} \in \mathbb{R}^{3\times h\times w} \mapsto \mathbf{X} \in \mathbb{R}^{3\times h\cdot \mathrm{s}_j \times w\cdot \mathrm{s}_j}$ represents a transformation by scaling) to the DINOv2 feature extractor $\mathcal{E}(\cdot)$ to get the attention maps from the final transformer block. Due to multi-headed attention, we get as many attention maps as the number of heads, which we average to get a single attention map per scale s_j as $A_j \in \mathbb{R}^{h_j \times w_j}$, where h_j, w_j are the scaled spatial dimensions.

Scoring Attention Maps. To find the optimal scale s^* for a given image crop X, we score the attention map A_j with a metric $\mathbf{S}(\cdot)$ that can quantify the *objectness* of the crop. To this end, we choose this metric to be Shannon entropy:

$$\mathbf{S}(A_j) = -\sum_{u,v} A_j(u,v) \log A_j(u,v), \tag{2}$$

where the attention map A_j is normalized to form a probability distribution over spatial locations (u, v). We then select the scale s_j that produces the lowest entropy attention map, as the optimal scale s^*:

$$s^* = \arg\min_{s_j \in \mathcal{S}} \mathbf{S}(A_j) \tag{3}$$

The rationale is that an attention map with lower entropy exhibits peaked activations, signaling the presence of detected objects. In contrast, a higher entropy indicates a

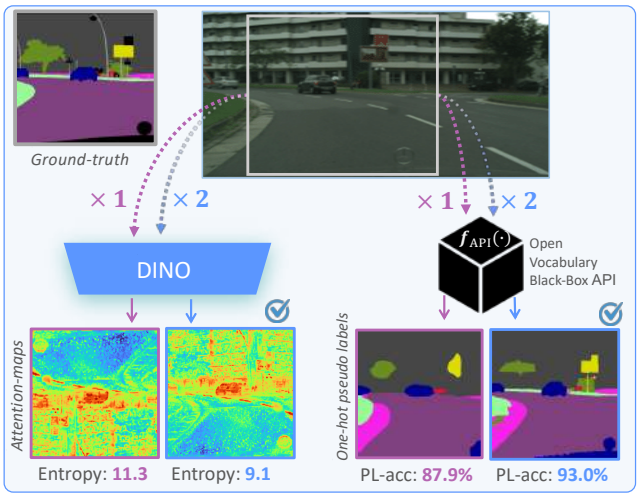


Figure 5. **DINOv2 and [CLS] Token for Optimal Resolution Selection.** DINOv2's [CLS] token attention maps are computed at multiple scales for each input image. The resolution with the highest spatially averaged attention score (*e.g.*, $\times 1$ vs. $\times 2$) is selected to generate pseudo-labels from the API model.

more diffuse attention map, suggesting that the model did not identify any particular semantic object. In Fig. 5 we visualize the attention-maps and the PLs and observe that a lower entropy attention map corresponds to better PL.

Discussion. A natural question arises: *Why should the student encoder serve as a good proxy for a black-box API model?* At first glance, this may seem counterintuitive, given that the student encoder and the API model are two separate and independently trained entities. Viewing through the lens of the Platonic Representation Hypothesis [30] – which suggests that as models scale, they tend to converge toward learning the same underlying features – we argue that both the API model and the student encoder (in our case, a pre-trained DINOv2), being trained on large-scale datasets, may develop internal representations that approximate the same statistical structures in representation space. Thus, both the models will react similarly to the same input, a property that we exploit in the absence of the access to the API's internal parameters.

4.2. Learning local segmentation model

Pseudo-labelling. We use the optimal scale s^* from Eq. (3) to scale the input image crop as $X^* = \mathcal{T}_{s^*}(X)$. We then feed X^* to the API model to get the PL, denoted by $\hat{Y} = \boldsymbol{f}_{\text{API}}(X^*, \mathcal{C})$, where the vocabulary $\mathcal{C}$ are converted into standard text prompts [14, 64], such as "`a photo of a [CLASS]`". More details are provided in Appendix B

Although ATGC ensures better PL quality than using random scales, some pixels can still have erroneous PLs, causing the student to overfit on noisy PLs. To further improve the quality of supervision, we perform another round of PL-filtering at the pixel-level. In detail, we compute the agreement in class predictions (or PL accuracy) between the API and the student model. The PLs provided by the API for a given image are used to supervise the student only when the PL accuracy exceeds a predefined threshold τ. We study the effect of varying τ in Appendix C.1.

Student training. We train the student model following Eq. (1) by setting $\alpha = 0$ and replacing Y with $\hat{Y}$, which is equivalent to a cross-entropy loss with hard PLs. To preserve its rich internal representation, we train only the linear decoder and the last transformer block of the student model. Further implementation details are provided in Appendix B.

5. Experiments

5.1. Experimental setup

Implementation. We conduct experiments on two unlabelled datasets following standard practices [28, 29, 61]. We use Cityscapes [16] with 2975 training and 500 validation images at 2048×1024 resolution. We also use ACDC [53] with 400 training and 200 validation images per weather condition (Night, Snow, Fog, Rain) at 1920×1080 resolution. We employ two open-vocabulary segmentation models as black-box models: SAN [64] with a ViT-L/14 backbone trained on COCO-Stuff [8], and CLIP-DINOiser, a training-free CLIP-based method with a ViT-B/16 backbone that has not been trained on segmentation data. For the local model, we use a DINOv2-S/14 encoder with registers [46] and a simple linear decoder.. We use thirteen different scale factors $\mathcal{S} = \{0.25, 0.28, 0.34, 0.38, 0.44, 0.47, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0\}$ for resolution selection. Additional training details are provided in Appendix B.

Evaluation Protocol and Baselines. We evaluate performance using mean Intersection over Union (mIoU) across 19 classes for all datasets. Since this work introduces a new B^2D setting, no established methods exist for direct comparison. We therefore compare ATGC against CoRTE [18], the closest available approach. Since the official CoRTE code was not available, we reimplemented it for our experiments. We also include several baselines: (i) "Naive Transfer" trains the student model using pseudo-labels from the API model at the default scale ($s_j = 1$); (ii) "Average" uses the average value of attention maps for scoring, with higher values preferred; (iii) "Random" randomly selects a pseudo-label corresponding to one of the available scales for each image crop; (iv) "Oracle" uses ground truth to evaluate all pseudo-labels at different scales and selects the one with highest pixel accuracy for training, providing an upper bound for our method; (v) "Supervised" is trained with real ground truth labels. Note that "Oracle" differs from "Supervised" : Oracle uses the best API-generated pseudo-labels (selected using ground truth), while "Supervised" uses actual ground truth labels for training.

Method	Logits	Road	SW	Buil.	Wall	Fence	Pole	TL	TS	Veg.	Ter.	Sky	PR	Rider	Car	Truck	Bus	Train	Mot.	Bike	mIoU
SAN [64]	-	88.7	50.1	82.0	29.2	33.4	3.1	22.5	29.8	81.3	20.9	86.1	53.5	0.7	73.1	23.9	60.0	50.3	46.8	53.0	46.8
CoRTE†[18]	✓	88.6	49.9	**82.4**	**29.9**	28.9	0.0	29.4	32.2	82.6	26.3	81.1	55.3	0.0	76.0	26.4	67.8	56.0	47.2	58.4	48.4
Naive Transfer	×	**89.2**	**50.4**	82.3	28.3	31.9	0.9	32.1	33.1	82.5	**28.8**	82.7	54.8	0.0	75.1	25.0	67.7	**56.4**	48.0	57.8	48.8
Average	×	88.4	45.6	80.5	25.0	27.7	0.3	25.9	28.2	80.4	21.6	81.5	51.9	0.0	74.3	24.3	62.4	29.6	44.7	55.6	44.6
Random	×	88.2	48.2	82.3	27.9	31.9	5.5	31.2	37.0	82.9	26.0	83.1	55.8	0.0	75.8	24.6	67.3	48.2	47.9	59.4	48.6
ATGC (Ours)	×	86.6	46.3	**82.4**	26.2	**35.3**	**10.5**	**34.6**	**41.7**	**83.8**	28.6	**84.7**	**58.0**	0.0	**76.6**	**23.8**	**67.9**	52.2	**50.8**	**61.3**	**50.1**
Oracle	-	91.8	56.8	84.4	40.7	41.7	9.4	35.6	41.3	84.8	35.0	83.7	57.6	0.0	79.2	28.4	70.0	60.8	50.8	61.1	53.3
Supervised	-	95.9	71.3	86.0	52.2	50.1	26.4	38.5	52.2	86.2	55.6	82.2	63.5	42.9	88.2	76.5	78.5	70.3	51.5	61.3	64.7

Table 2. **Specialization to Cityscapes with SAN.** The best score for each column is highlighted in **bold**. Results are obtained using prompt engineering and averaged over 3 random seeds.† Reimplemented by us as the original code was not available.

5.2. Main results

Specialization to Cityscapes. We evaluate ATGC using two different open-vocabulary models as APIs: SAN [64] and CLIP-DINOiser [62]. Tabs. 2 and 3 show the results for both settings, respectively. For SAN as the API model, our method achieves 50.1 mIoU, outperforming all baselines including CoRTE† (48.4), Naive Transfer (48.8), Random (48.6), and Average (44.6). With CLIP-DINOiser as the API, our method reaches 37.9 mIoU, again surpassing CoRTE† (34.5), Naive Transfer (34.3), Average (34.1), and Random (35.5). Notably, our ATGC operates *without* access to API logits, while CoRTE† requires them, making our setting more realistic. Although our method falls short of Oracle performance (53.3 and 40.9 mIoU), it consistently outperforms existing approaches. The Oracle represents an upper bound achieved by always selecting optimal pseudo-labels, while the Supervised baseline (64.7 mIoU) shows potential performance when training with ground truth labels. These results validate our scale selection strategy's effectiveness for black-box model distillation.

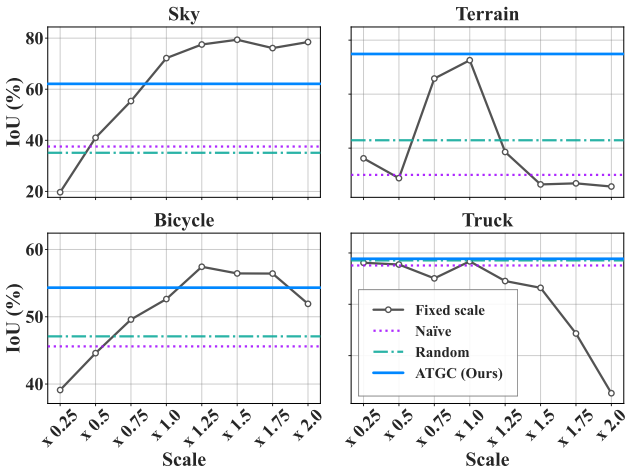


Figure 6. **Class-wise IoU performance.** The fixed scale approach (**gray** line) shows performance when training at specific scales. At the same time, horizontal baselines represent **naive** (no scaling), **random** scale selection, and our **ATGC** method. Results demonstrate that optimal scales vary significantly across classes, with ATGC consistently achieving competitive or superior performance compared to baseline approaches.

Specialization to ACDC. Tab. 4 demonstrates ATGC's superior performance across both evaluation settings and black-box API models (SAN and CLIP-DINOiser). In the standard setting where models train on unlabelled ACDC images, ATGC achieves 41.0 mIoU with SAN and 25.0 mIoU with CLIP-DINOiser, consistently outperforming all baselines. Remarkably, in the domain generalization scenario, models trained on unlabelled Cityscapes and evaluated on ACDC achieve superior performance compared to direct ACDC training, a phenomenon we examine in 3.

5.3. In-depth analysis

1 **Does ATGC always find the "optimal" scales?.** To investigate this, we perform a grid search over several scales, where each scale is used to train a separate student model for the entire training duration. We then compare the performance of these fixed-scale models against ATGC. As reported in Fig. 6, we observe that for the "sky" class, the performance of a student trained using PLs from a fixed scale of $\times 1.5$ surpasses that of ATGC. Similarly, class "bicycle" benefits from scale $\times 1.25$, a scale that ATGC fails to always find. In contrast, for certain classes such as "terrain" and "truck", ATGC achieves comparable or even superior results compared with the grid search scales. These results indicate that ATGC does not universally find the optimal scales for all the classes, which could be attributed to the presence of multiple competing classes within an image crop. However, ATGC is more compute-efficient and grid search is infeasible in practice due to the lack of any labelled data.

2 **How does the API model's training data affect performance?.** The quality of pseudo-labels, and thus student performance, is greatly dependent on the domain gap between the API model's training data and the target domain [52]. This is evidenced by the performance gap between the original CoRTE results (55.5 mIoU on Cityscapes) [18], which used an API model trained on the closely-aligned GTA dataset, and our CoRTE† baseline using SAN as an open-vocabulary API (48.4 in Tab. 2). This discrepancy arises because Cityscapes shares more similarities in scene layout and semantic classes with GTA than

Method	Logits	Road	SW	Buil.	Wall	Fence	Pole	TL	TS	Veg.	Ter.	Sky	PR	Rider	Car	Truck	Bus	Train	Mot.	Bike	mIoU
CLIP-DINOiser [62]	-	68.5	17.0	73.7	23.4	24.0	10.1	2.2	3.5	78.8	10.2	43.3	45.2	9.2	69.3	26.1	52.9	26.8	27.4	42.3	34.4
CoRTE† [18]	✓	**64.9**	18.5	75.8	23.1	18.5	10.0	0.0	0.0	80.5	4.5	36.9	52.0	3.0	72.9	30.2	57.1	24.6	**35.8**	47.2	34.5
Naive Transfer	×	64.0	17.8	75.2	21.4	21.4	10.2	0.0	0.0	80.6	5.0	37.6	49.7	**4.0**	71.3	27.6	56.0	28.5	35.0	45.6	34.3
Average	×	65.5	18.3	72.2	**24.3**	**25.3**	8.4	0.0	0.0	76.4	8.6	25.6	48.8	2.4	67.9	27.2	55.9	44.5	32.5	43.5	34.1
Random	×	64.8	**18.7**	74.5	23.9	24.3	10.3	0.0	0.0	79.1	11.5	35.6	51.2	1.6	70.6	28.2	**58.2**	**41.4**	33.3	47.3	35.5
ATGC (Ours)	×	47.7	15.6	**78.0**	22.5	24.2	**16.3**	**0.2**	**4.7**	**83.1**	**27.4**	**62.1**	**53.0**	0.7	**73.1**	**28.9**	56.1	35.8	35.5	**54.3**	**37.9**
Oracle	-	76.7	24.4	78.8	30.5	31.3	14.2	0.0	2.5	81.5	25.4	49.3	52.8	6.5	74.1	33.4	62.3	48.6	34.1	49.9	40.9
Supervised	-	95.9	71.3	86.0	52.2	50.1	26.4	38.5	52.2	86.2	55.6	82.2	63.5	42.9	88.2	76.5	78.5	70.3	51.5	61.3	64.7

Table 3. **Specialization to Cityscapes with CLIP-DINOiser.** The best score for each column is highlighted in **bold**. Results are obtained using prompt engineering and averaged over 3 random seeds.† Reimplemented by us as the original code was not available.

	Method	rain	fog	night	snow	Average
SAN	SAN [64]	45.5	45.9	31.8	48.0	42.8
	CoRTE [18]	42.7	**45.7**	31.9	42.0	40.6
	Naive Transfer	40.9	45.3	32.0	41.2	39.9
	Random	41.1	45.2	32.5	41.0	40.0
	ATGC	**42.9**	45.1	**33.6**	**42.3**	**41.0**
	Oracle	45.2	49.1	36.6	44.8	44.0
	Domain Generalization Evaluation					
	Naive Transfer (DG)	46.1	47.2	33.3	47.0	43.4
	Random (DG)	45.9	48.0	33.3	47.8	43.8
	ATGC (DG)	**48.0**	**49.4**	**34.3**	**48.5**	**45.1**
CLIP-DINOiser	CLIP-DINOiser [62]	32.0	31.6	13.7	30.4	26.9
	CoRTE [18]	24.9	24.5	10.8	26.6	21.7
	Naive Transfer	25.1	21.6	11.1	22.4	20.1
	Random	26.1	24.0	12.5	23.4	21.5
	ATGC	**27.7**	**27.9**	**17.2**	**27.0**	**25.0**
	Oracle	34.6	38.5	23.1	34.7	32.7
	Domain Generalization Evaluation					
	Naive Transfer (DG)	29.4	31.4	18.2	31.7	27.7
	Random (DG)	32.0	32.6	19.7	32.9	29.3
	ATGC (DG)	**36.8**	**37.2**	**23.6**	**39.7**	**34.3**

Table 4. **Specialization to ACDC.** Results of training **ATGC** on the unlabelled ACDC dataset compared to state-of-the-art specialization methods across varying weather conditions. The best score for each column is highlighted in **bold**.

with COCO-Stuff, on which SAN was trained. The performance drop is even more pronounced with CLIP-DINOiser as the API, where CoRTE†'s performance decreases to 34.5 in Tab. 3, as it is a training-free model that has not been exposed to any segmentation dataset. We argue that achieving high performance as in CoRTE's original setting can be illusory, as finding a target-aligned API model is often difficult in practice, particularly since the training data of commercial APIs is typically not disclosed. Therefore, assuming the API model is a generalist open-vocabulary model that covers a wide variety of target domains aligns better with real-world constraints.

3 **Does dataset size play a role?** We investigate why the DG results reported in Tab. 4 consistently outperform direct training on ACDC, an observation that is counterintuitive in the DG literature [17]. We believe that this difference is attributed to the smaller size of the ACDC dataset (2,975 in Cityscapes versus 1,600 in ACDC), as also shown in [37]. Other factors could be at play, such as weather-corrupted images in ACDC, which are typically long-tailed in open-vocabulary pre-training datasets, resulting in noisier PLs. Thus, we conclude that it is important to have a sufficiently large dataset for effective B^2D, and caution must be exercised when working with target domains that are characterized by heavy corruptions or very long-tailed distributions.

6. Conclusions and Limitations

In this work, we introduced Black-Box Distillation in semantic segmentation (B^2D), a realistic adaptation paradigm that operates with only one-hot predictions from open-vocabulary API models, removing impractical assumptions of existing black-box approaches. We identified the "curse of resolution", whereby different object classes achieve optimal segmentation at different input scales, and proposed ATGC (**AT**tention-**G**uided s**C**aler), which leverages DINOv2 attention maps and entropy scoring to dynamically select optimal scales for black-box inference, demonstrating effectiveness in generating high-quality pseudo-labels.

Although our experiments show promising results, several limitations remain. As discussed in Sec. 5.3, ATGC operates at the image level rather than the class level and uses static prompts that do not adapt to varying object scales. This can produce noisier pseudo-labels when multiple classes benefiting from different scales appear in the same crop. Future work could address this by introducing class-level scaling or dynamic prompts. Additionally, our current approach does not account for API call budgets; efficiency could be improved by focusing on the most informative regions of an image to accelerate convergence.

Acknowledgements

This work has been supported by the French National Research Agency (ANR) with the ANR-20-CE23-0027 project and was granted access to the HPC resources of IDRIS under the allocations AD011013071 and AD011014921 made by GENCI. We also thank Nacereddine Laddaoui for his invaluable help with infrastructure.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1, 3

[2] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. In *ECCV*, 2022. 5

[3] Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundation models defining a new era in vision: a survey and outlook. *PAMI*, 2025. 3

[4] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024. 1

[5] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *CVPR*, pages 10925–10934, 2022. 3

[6] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1, 3

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 1

[8] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018. 6

[9] Wei-Lun Chang, Hui-Po Wang, Wen-Hsiao Peng, and Wei-Chen Chiu. All about structure: Adapting structural information across domains for boosting semantic segmentation. In *CVPR*, pages 1900–1909, 2019. 3

[10] Soumitri Chattopadhyay, Basar Demir, and Marc Niethammer. Zero-shot domain generalization of foundational models for 3d medical image segmentation: An experimental study. *arXiv preprint arXiv:2503.22862*, 2025. 3

[11] Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023. 1

[12] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 40(4):834–848, 2017. 2

[13] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023. 1

[14] Seokju Cho, Heeseong Shin, Sunghwan Hong, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. In *CVPR*, 2024. 6

[15] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 1

[16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 6, 3

[17] Gabriela Csurka, Riccardo Volpi, Boris Chidlovskii, et al. Semantic image segmentation: Two decades of research. *Foundations and Trends® in Computer Graphics and Vision*, 14(1-2):1–162, 2022. 8

[18] Claudia Cuttano, Antonio Tavera, Fabio Cermelli, Giuseppe Averta, and Barbara Caputo. Cross-domain transfer learning with corte: Consistent and reliable transfer from black-box to lightweight segmentation model. In *CVPR*, 2023. 2, 3, 4, 6, 7, 8, 5

[19] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[20] Francois Fleuret et al. Uncertainty reduction for model adaptation in semantic segmentation. In *CVPR*, pages 9613–9623, 2021. 3

[21] Greg Heinrich, Mike Ranzinger, Hongxu Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. Radiov2. 5: Improved baselines for agglomerative vision foundation models. In *CVPR*, pages 22487–22497, 2025. 3

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1

[23] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 3, 4

[24] Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*, 2022. 2

[25] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and

Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, pages 1989–1998, 2018. 3

[26] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016. 1, 3

[27] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *CVPR*, pages 1335–1344, 2018. 3

[28] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*, pages 9924–9935, 2022. 4, 6

[29] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. In *ECCV*, 2022. 2, 4, 6

[30] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024. 6

[31] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991. 1

[32] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics*, 2020. 2

[33] Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcet, Hu Xu, Daniel Li, Marc Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, et al. Dinov2 meets text: A unified framework for image-and pixel-level vision-language alignment. In *CVPR*, pages 24905–24916, 2025. 2, 3

[34] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic gradient descent. In *ICLR: international conference on learning representations*, pages 1–15, 2015. 1

[35] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. *ICCV*, 2023. 1, 3

[36] Jogendra Nath Kundu, Akshay Kulkarni, Amit Singh, Varun Jampani, and R Venkatesh Babu. Generalize then adapt: Source-free domain adaptive semantic segmentation. In *ICCV*, pages 7046–7056, 2021. 3

[37] Giulia Lanzillotta, Felix Sarnthein, Gil Kur, Thomas Hofmann, and Bobby He. Testing knowledge distillation theories with dataset size. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024. 8

[38] Jian Liang, Dapeng Hu, Jiashi Feng, and Ran He. Dine: Domain adaptation from single and multiple black-box predictors. In *CVPR*, 2022. 3

[39] Bingbin Liu, Sebastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. Tinygsm: achieving 80% on gsm8k with small language models. *arXiv preprint arXiv:2312.09241*, 2023. 2

[40] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2024. 1

[41] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *NeurIPS*, 33:20331–20342, 2020. 5

[42] Yuang Liu, Wei Zhang, and Jun Wang. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing*, 415:106–113, 2020. 3

[43] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *CVPR*, 2021. 3

[44] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *AAAI*, pages 5191–5198, 2020. 3

[45] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*, 2023. 2

[46] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 2, 3, 5, 6

[47] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. *arXiv preprint arXiv:2004.07703*, 2020. 3

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PmLR, 2021. 3, 1

[49] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *CVPR*, pages 12490–12500, 2024. 3

[50] Raspberry Pi Foundation. Raspberry Pi 4 Model B. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/, 2019. Accessed: 2025-07-16. 3

[51] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 102–118. Springer, 2016. 4

[52] Subhankar Roy, Evgeny Krivosheev, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Curriculum graph co-teaching for multi-target domain adaptation. In *CVPR*, pages 5351–5360, 2021. 7

[53] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *CVPR*, 2021. 2, 6

[54] Mert Bülent Sarıyıldız, Philippe Weinzaepfel, Thomas Lucas, Diane Larlus, and Yannis Kalantidis. Unic: Universal classification models via multi-teacher distillation. In *ECCV*, pages 353–371. Springer, 2024. 3

[55] Haotian Sun, Yuchen Zhuang, Wei Wei, Chao Zhang, and Bo Dai. Bbox-adapter: Lightweight adapting for black-box large language models. *arXiv preprint arXiv:2402.08219*, 2024. 2, 3

[56] Ximeng Sun, Pengchuan Zhang, Peizhao Zhang, Hardik Shah, Kate Saenko, and Xide Xia. Dime-fm: Distilling multimodal and efficient foundation models. In *ICCV*, pages 15521–15533, 2023. 1, 3, 4

[57] Yi Tay, Mostafa Dehghani, Samira Abnar, Hyung Chung, William Fedus, Jinfeng Rao, Sharan Narang, Vinh Tran, Dani Yogatama, and Donald Metzler. Scaling laws vs model architectures: How does inductive bias influence scaling? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023. 1

[58] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

[59] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 3

[60] Yixuan Wei, Han Hu, Zhenda Xie, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. Contrastive learning rivals masked image modeling in fine-tuning via feature distillation. *arXiv preprint arXiv:2205.14141*, 2022. 3

[61] Zhixiang Wei, Lin Chen, Yi Jin, Xiaoxiao Ma, Tianle Liu, Pengyang Ling, Ben Wang, Huaian Chen, and Jinjin Zheng. Stronger fewer & superior: Harnessing vision foundation models for domain generalized semantic segmentation. In *CVPR*, 2024. 6

[62] Monika Wysoczańska, Oriane Siméoni, Michaël Ramamonjisoa, Andrei Bursuc, Tomasz Trzciński, and Patrick Pérez. Clip-dinoiser: Teaching clip a few dino tricks for open-vocabulary semantic segmentation. In *ECCV*, pages 320–337. Springer, 2024. 7, 8, 1

[63] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, 2023. 2

[64] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *CVPR*, 2023. 4, 6, 7, 8, 1

[65] Chuanguang Yang, Zhulin An, Linhang Cai, and Yongjun Xu. Knowledge distillation using hierarchical self-supervision augmented distribution. *IEEE transactions on neural networks and learning systems*, 35(2):2094–2108, 2022. 2

[66] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, pages 4133–4141, 2017. 3

[67] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1285–1294, 2017. 3

[68] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. In *CVPR*, 2024. 1

[69] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 3

[70] Haojian Zhang, Yabin Zhang, Kui Jia, and Lei Zhang. Unsupervised domain adaptation of black-box source models. *arXiv preprint arXiv:2101.02839*, 2021. 2, 3

[71] Jingyi Zhang, Jiaxing Huang, Xueying Jiang, and Shijian Lu. Black-box unsupervised domain adaptation with bi-directional atkinson-shiffrin memory. In *ICCV*, 2023. 3

[72] Yuanbing Zhu, Bingke Zhu, Zhen Chen, Huan Xu, Ming Tang, and Jinqiao Wang. Mrovseg: Breaking the resolution curse of vision-language models in open-vocabulary semantic segmentation. *arXiv preprint arXiv:2408.14776*, 2024. 4

Make me an Expert: Distilling from Generalist Black-Box Models into Specialized Models for Semantic Segmentation

Supplementary Material

In this supplementary material, we provide comprehensive details about the experimental results from the main paper, including additional qualitative results. The supplementary material is organized as follows: In Section A, we present the pseudo-code for Attention Maps construction. In Section B, we provide additional details about the training process of ATGC. Section C presents detailed experimental results and some failure case analysis. Finally, in Section D we discuss the limitations and future directions in Black-Box Distillation (B^{2}D).

A. Pseudo-Code

We present the pseudo-code for ATGC in Algorithm 1 and Algorithm 2. Algorithm 1 details the offline construction of Attention Maps for the entire dataset, while Algorithm 2 describes the training procedure of ATGC that leverages these attention maps for resolution mining and pseudo-label generation. Note that in our implementation, we construct the attention maps offline, and do it only once for the entire dataset because it does not depend on the API model. Offline computation speeds up the training process, as multiple forward passes with varying crop resolutions are not needed to mine the optimal resolution.

Algorithm 1 Attention Maps Construction

Require: Dataset $\mathcal{D} = \{X_i\}_{i=1}^{n}$ of size n, Scale factors $\mathcal{S} = \{\mathrm{s}_j \mid j \in \mathbb{N}, \mathrm{s}_{min} \leq \mathrm{s}_j \leq \mathrm{s}_{max}\}$
Ensure: Dataset of Attention Maps $\{A_{i,j}\}_{i=1,j}^{n,|\mathcal{S}|}$
for each image X_i in $\mathcal{D}$ **do**
  for each scale factor s_j in $\mathcal{S}$ **do**
    $X_{i,j} \leftarrow \mathcal{T}_{\mathrm{s}_j}(X_i)$ ▷ Scale image by factor s_j
    $A_{i,j} \leftarrow \mathcal{E}(X_{i,j})$ ▷ Get DINOv2 attention map
    $A_{i,j} \leftarrow \mathcal{T}_{1/\mathrm{s}_j}(A_{i,j})$ ▷ Rescale to original dimensions
  end for
end for
return Dataset of attention maps $\{A_{i,j}\}_{i=1,j}^{n,|\mathcal{S}|}$

B. Implementation details

We train our model with the AdamW optimizer [34] using a weight decay of 0.05 and a learning rate of 10^{-5} with polynomial decay. We set training iterations to 10K for Cityscapes and 6K for ACDC, with a batch size of 4. Following standard evaluation protocols, the student model is evaluated on resized validation images to maintain consistent inference conditions across datasets, specifically resizing Cityscapes images from 2048×1024 to 1024×512, ACDC images from 1920×1080 to 960×540 for all weather conditions (Night, Snow, Fog, Rain). All reported performance metrics reflect the student model's capabilities after the full training schedule, ensuring fair comparisons with existing methods and maintaining computational efficiency.

Algorithm 2 Training Procedure with ATGC

Require: Dataset $\mathcal{D}$, Dataset of Attention Maps $\{A_{i,j}\}_{i=1,j}^{n,|\mathcal{S}|}$, Class names $\mathcal{C}$, API model $\boldsymbol{f}_{\text{API}}$, Student network $\mathcal{F}_\theta$, Threshold τ
for each iteration t **do**
  Sample image X from $\mathcal{D}$
  $X_c, A_c \leftarrow \texttt{RandomCrop}(X, A)$ ▷ Get fixed-size crops
  $s^* \leftarrow \text{argmin}_j\, \mathbf{S}(A_{c,j})$ ▷ Find optimal scale with lowest entropy
  $X_c^* \leftarrow \mathcal{T}_{s^*}(X_c)$ ▷ Scale image by scale factor s^*
  $\hat{Y} \leftarrow \boldsymbol{f}_{\text{API}}(X_c^*, \mathcal{C})$ ▷ Get API pseudo-labels
  $\hat{Y} \leftarrow \mathcal{T}_{1/s^*}(\hat{Y})$ ▷ Rescale pseudo-labels to original crop size
  $\tilde{Y} \leftarrow \mathcal{F}_\theta(X_c)$ ▷ Get student prediction
  IoU $\leftarrow \texttt{IntersectionOverUnion}(\hat{Y}, \tilde{Y})$ ▷ Compute consistency
  if IoU $\geq \tau$ **then**
    Update θ using Eq. (1) with $\hat{Y}$ as supervision
  end if
end for
return Updated student model $\mathcal{F}_\theta$

Prompt engineering. In this work, we have considered two open-vocabulary segmentation models –CLIP-DINOiser [62] and SAN [64] – as the API model. Following standard practice in open-vocabulary segmentation literature, we evaluate on 19 semantic classes from the Cityscapes dataset, using both primary class names and their synonyms to enhance model robustness (see class list below). For prompting the open-vocabulary models, we used the commonly used 80 templates from CLIP for zero-shot image classification, commonly referred to as ImageNet templates [48]. We computed the average of the text embeddings from all templates to obtain the final text embedding for each class during inference. **Classes:** ["road, street, highway", "sidewalk, pavement, footpath", "build-

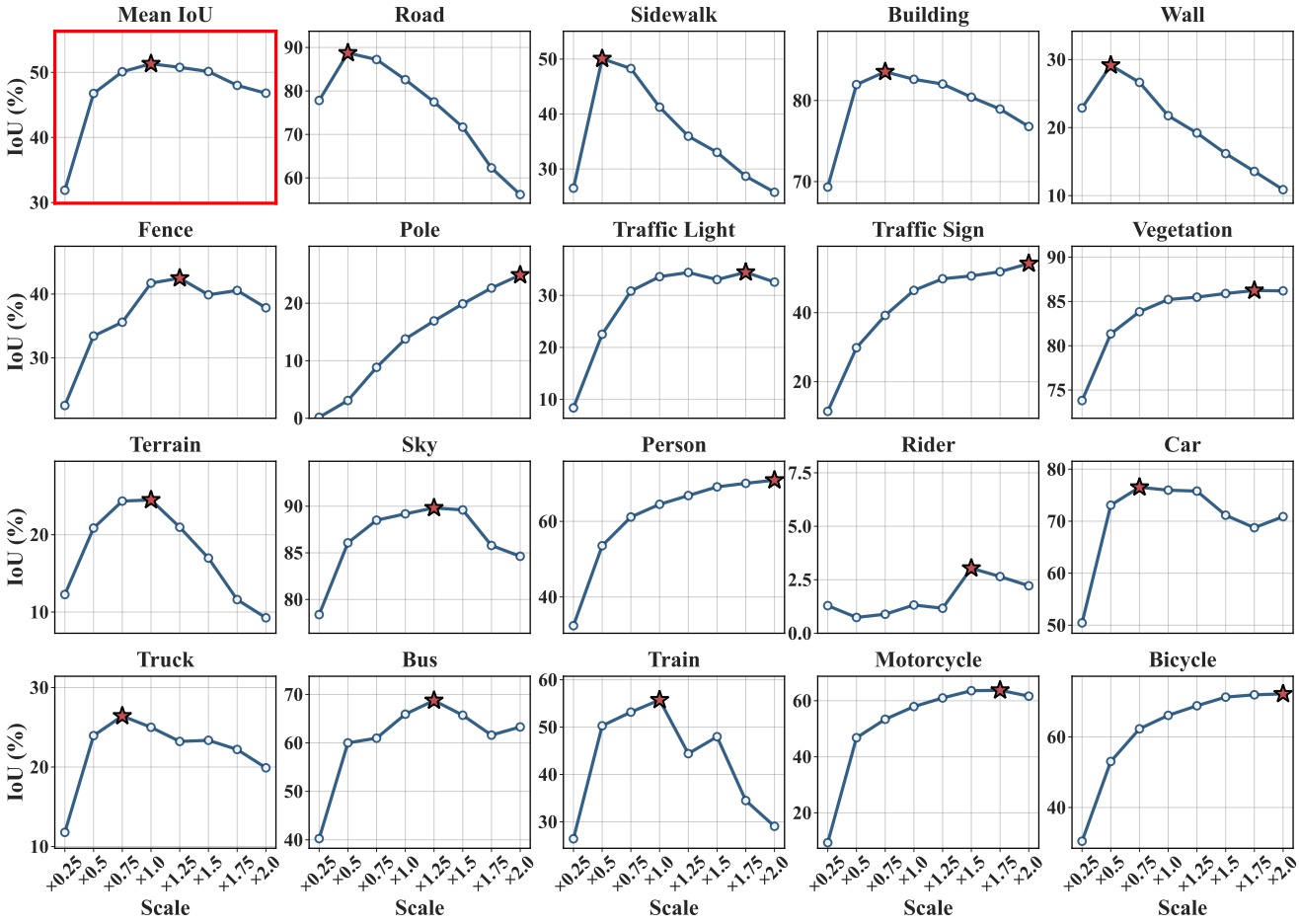


Figure 7. **Performance of SAN across all scaling factors.** The first subplot (top-left) shows the impact of resolution scaling on average mIoU, with subsequent subplots detailing individual class performances. Scaling factors ($\times 0.25$, $\times 1$, ..., $\times 2$) represent resolution changes, yielding insights that highlight the optimal resolution for each class. Markers represent peak performance, highlighting the optimal resolution for each class. Small objects like "pole" and "bicycle" are better segmented in very high-resolution images which result in detailed crops given to the API model, while semantic classes such as "road" and "wall" are better segmented in low-resolution images yielding large context crops fed to the API model.

ing, structure, house", "wall, brick wall, stone wall", "fence, barrier, hedge", "pole, post, pillar", "traffic light, red light, green light", "traffic sign, stop sign, warning sign", "vegetation, plants, trees", "terrain, ground, grass", "sky, air, clouds", "person, pedestrian, people", "rider, biker, driver", "car, automobile, vehicle", "truck, pickup, van", "bus, shuttle, minibus", "train, tram, locomotive", "motorcycle, motorbike, scooter", "bicycle, bike, cycle",]

C. Experimental results

C.1. Pseudo-label filtering.

We ablate the pseudo-label filtering threshold $\tau \in [0.0, 0.9]$ to study its impact on performance when using CLIP-DINOiser as the API model. As shown in Fig. 11, ATGC achieves optimal performance at $\tau = 0.7$ with 37.9% mIoU. Notably, ATGC demonstrates robust performance even when pseudo-label filtering is completely deactivated ($\tau = 0.0$), achieving 37.2% mIoU with only a modest 0.7% decrease compared to the optimal threshold. This indicates that ATGC works effectively regardless of whether PL filtering is activated or not, showcasing the inherent quality of our scale-optimized pseudo-labels. Performance degrades more significantly when τ is too restrictive ($\tau = 0.9$: 31.5%), as overly strict filtering removes valuable training signals, reducing effective pseudo-label coverage. The optimal threshold $\tau = 0.7$ achieves the right balance between pseudo-label quality and quantity. For consistency, we report all main results using $\tau = 0.7$ throughout our experiments.

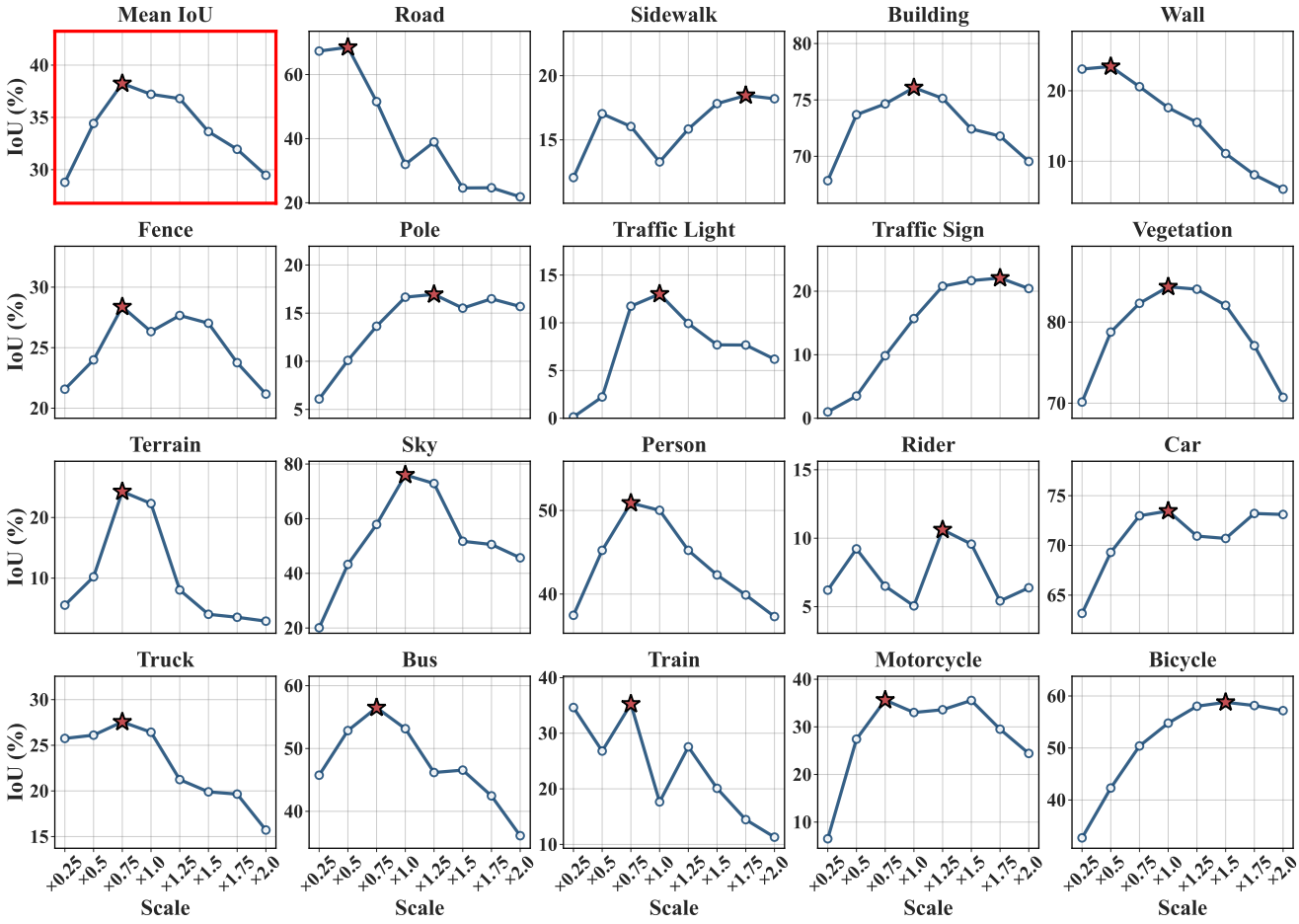


Figure 8. **Performance of CLIP-DINOiser across all scaling factors.** The first subplot (top-left) shows the impact of resolution scaling on average mIoU, with subsequent subplots detailing individual class performances. Scaling factors ($\times 0.25$, $\times 1$, ..., $\times 2$) represent resolution changes, yielding insights that highlight the optimal resolution for each class. Markers represent peak performance, highlighting the optimal resolution for each class.

C.2. Effect of varying resolutions

Figures 7 and 8 show the performance of SAN and CLIP-DINOiser, respectively, when used as API models, across various scaling factors applied to the initial image resolution (1024×2048) of the Cityscapes [16] validation set. The subplots illustrate the effect of image resolution on segmentation accuracy for each semantic class and the average mIoU. Each subplot highlights scaling factors ranging from 0.25 to 2.0, with blue star indicating the optimal resolution for each class.

The first subplot shows that the average mIoU across all classes peaks at the scaling factor of 1.0, reflecting the best overall performance. However, peak performance varies across classes at different scale factors. For example, classes such as "person" and "motorcycle" perform optimally at scaling factors of 2.0 and 1.5, respectively. In contrast, "sidewalk" and "truck" are better segmented at scaling factors of 0.5 and 0.75, respectively.

In general, the results emphasize that resolution scaling plays a critical role in achieving optimal performance for each class. Optimal resolution choices vary across different object types, with some benefiting from detailed crops taken from high resolution base images and others from large context crops taken from low resolution base images. This suggests that a tailored resolution strategy can improve segmentation outcomes using the API model, highlighting the importance of careful resolution scaling rather than adopting a one-size-fits-all approach.

C.3. Qualitative comparison of methods

In Fig. 10, we present the qualitative comparison of different methods. We have compared our proposed ATGC with a B^2UDA approach CoRTe [18], and the Naive Transfer baseline that directly distills from the API model without considering varying resolutions. From the figure we observe that

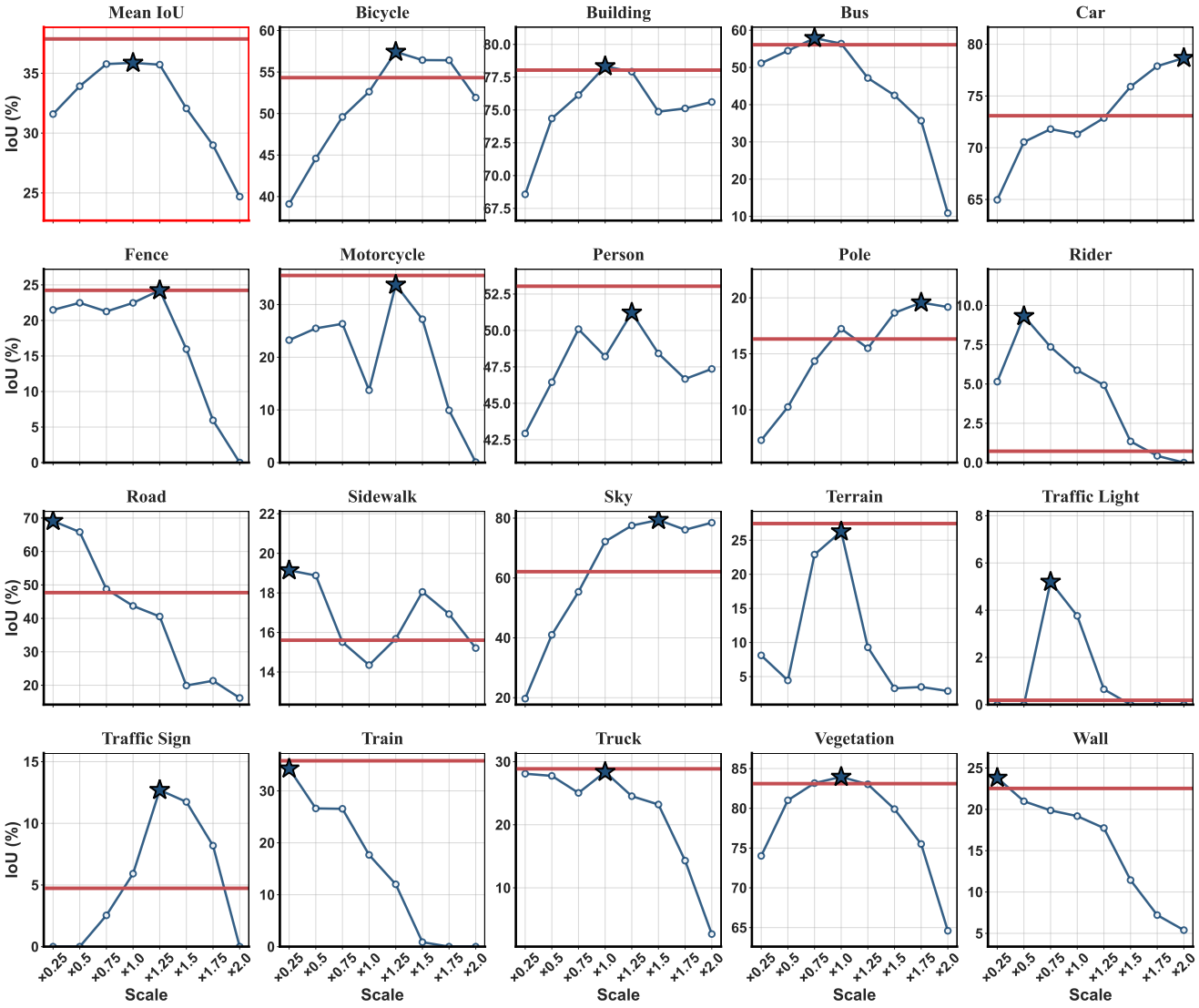


Figure 9. **Class-wise performance after training across all scaling factors.** The first subplot (top-left) shows the impact **training with a fixed resolution** on average mIoU, with subsequent subplots detailing individual class performances. The red line represents our method ATGC, and the blue curve shows results from training with fixed scaling factors. Each scaling factor ($\times 0.25, \times 1, ..., \times 2$) indicates the scale applied to image crops during training. Blue markers highlight peak performance points, indicating the optimal scaling factor for each semantic class.

for larger object classes, such as "car" and "bus", and stuff classes, such as "road" and "vegetation" the segmentation quality is reasonably fair among all the baselines and our ATGC. However, when it comes to smaller objects, such as "poles" (highlighted with a white box in all the segmentation maps), the segmentation quality of our ATGC is much better. Furthermore, in row 3 of Fig. 10 we observe in the highlighted region of the segmentation map that CoRTe is worse at segmenting "sidewalk" that is farther in the scene, compared to the nearby "sidewalk". Contrarily, both the Naive Transfer and ATGC is better at segmenting farther objects, with ATGC doing well on the farthest "pole" in the image. This observation underscores the need of obtaining pseudo-labels at the optimal crop resolution to reasonably segment objects that occupy a very small portion of the image.

C.4. Effectiveness of Attention Maps

Building upon the motivational experiment in Sec. 4 and Fig. 4 of the main paper, which demonstrated a positive correlation between attention map quality (computed between DINO [CLS] and patch tokens) and segmentation

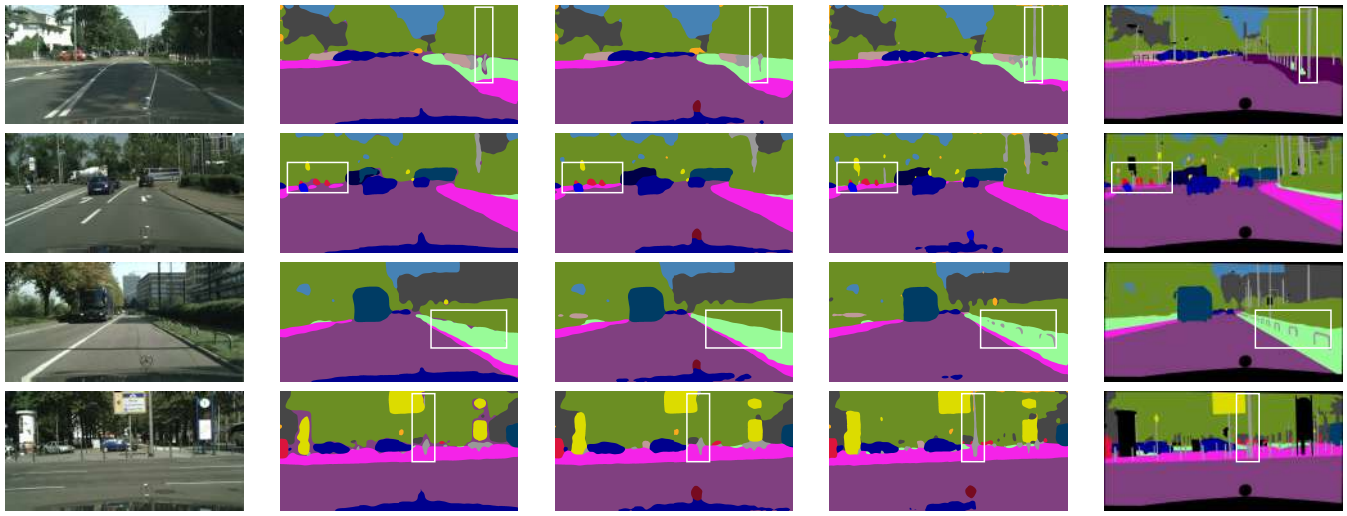


Figure 10. **Qualitative comparison of different methods.** From left to right: input RGB image, predictions from CoRTE [18], Naive Transfer, our method ATGC (Ours), and ground-truth segmentation maps. Our method shows improved segmentation quality of small objects, such as "poles" (highlighted with white rectangular boxes), compared to the other approaches.

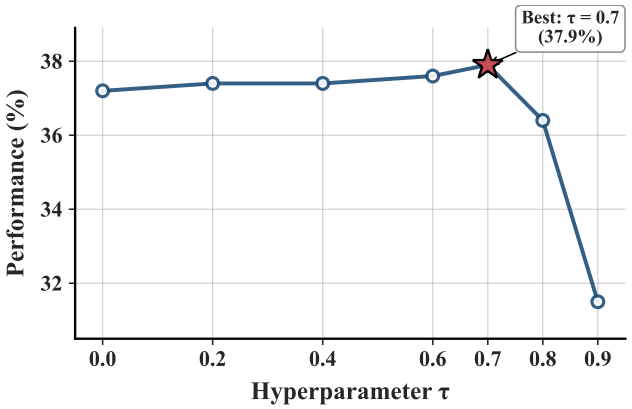


Figure 11. **Effect of confidence threshold τ on pseudo-label filtering performance.**

performance, we present additional qualitative validation in Fig. 12.

These examples confirm that the relationship between attention map entropy and pseudo-label (PL) quality is consistent across different scenarios. Specifically, attention maps with lower entropy display concentrated, peaked activations that correspond to better object localization and, consequently, higher-quality pseudo-labels. Conversely, higher entropy values indicate spatially diffuse attention patterns that correlate with poor-quality pseudo-labels. Our experimental analysis validates that Shannon entropy of attention maps serves as an effective and reliable proxy for identifying the optimal scale for pseudo-label generation, which forms the core mechanism our method uses to distill knowledge from the API model.

C.5. Failure cases

In Fig. 13, we report some failure modes of our ATGC, where we show that optimal crop resolution may not always result in the best pseudo-label predictions from the API model. In detail, for a given random crop, we visualize the prediction of the API model at the original crop resolution ($\boldsymbol{x}_c$) and the one with the optimal scale resolution ($\boldsymbol{x}_c^*$) as determined by the ATGC module. We observe from the figure that the quality of segmentation maps predicted by the API at the optimal scale is inferior when compared with the one obtained with the original crop resolution. Specifically, from Fig. 13(c) and (d) we observe that the optimal crop resolution introduces strange artifacts in the segmentation maps, *e.g.*, sharp discontinuities. Similar kind of observation holds for the other figures, where incorrect classes are predicted within other objects (*e.g.*, Fig. 13(a), where a "truck" pixels encroaches onto a "car" class pixels). This phenomenon occurs when extreme *zoom-in* operations (with high scale factors) cause attention maps to be computed over homogeneous image regions, leading to highly concentrated activations for a single class and consequently producing misleadingly low entropy values.

Despite these pathological predictions, we employ a PL-filtering technique (as discussed in Sec. 4 of the main paper) that uses the consistency between the API predictions and the student model predictions to avoid training on erroneous PLs. Since the student model processes the image crop at original resolution, it is less likely to make the same errors as the API model that always generates PLs at the optimal

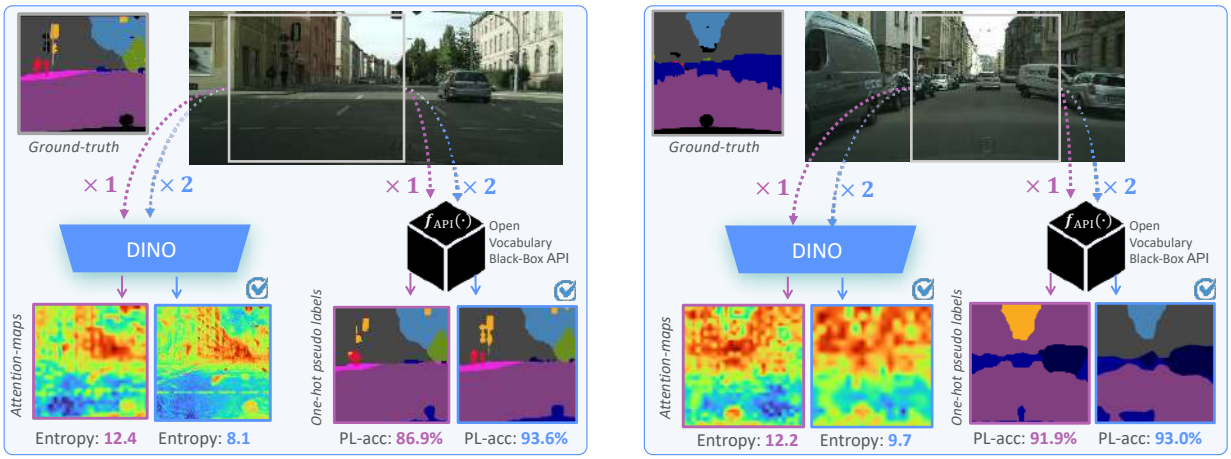
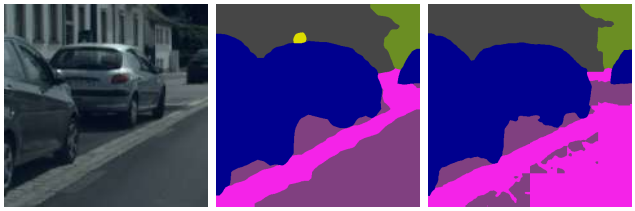


Figure 12. **DINOv2 and [CLS] Token for Optimal Resolution Selection.** DINOv2's [CLS] token attention maps are computed at multiple scales for each input image. The resolution with the highest spatially averaged attention score (*e.g.*, $\times 1$, $\times 1.5$ and $\times 2$) is selected to generate pseudo-labels from the API model.

Example 1: (left) RGB input, (middle) prediction on $\mathbf{X}_c$, (right) prediction on $\mathbf{X}_c^*$



Example 2: (left) RGB input, (middle) prediction on $\mathbf{X}_c$, (right) prediction on $\mathbf{X}_c^*$

Example 3: (left) RGB input, (middle) prediction on $\mathbf{X}_c$, (right) prediction on $\mathbf{X}_c^*$

Example 4: (left) RGB input, (middle) prediction on $\mathbf{X}_c$, (right) prediction on $\mathbf{X}_c^*$

Figure 13. **Failure cases of our framework**. The figure illustrates cases where the pseudo-label given by the API model on the cropped image ($\mathbf{X}_c$) is more accurate than the one given by the API using the optimal scale ($\mathbf{X}_c^*$). Despite leveraging the optimal scale, the API may struggle to accurately segment certain classes in some specific scenarios, like the ones depicted in this figure. However, these pseudo-labels get filtered during training using the consistency measure with the predictions given by the student model.

scale resolution (see Fig. 3 of the main paper). We compute the pixel-level agreement between the API pseudo-labels and the student predictions using pixel accuracy, and only use pseudo-labels for supervision when this agreement exceeds a predefined threshold τ. This filtering mechanism ensures that pathological API predictions are discarded and do not negatively impact the student network training.

D. Limitations and future works

In this work, we introduced the task of B^2D, which distills knowledge from a generalist black-box model into a local segmentation model by mining optimal scales for querying the API. This approach represents a significant improvement over using default or random crop resolutions for pseudo-label generation. However, our current framework operates under the assumption that optimal scale selection is the primary determinant of pseudo-label quality. While

our experiments validate that optimal scaling plays a crucial role in determining PL quality, several other factors may influence performance that remain unexplored, such as adaptive prompt engineering tailored to specific crop resolutions. Additionally, our framework lacks a systematic mechanism for selecting the most informative samples or image regions for API queries. This uniform treatment of all image regions is inefficient, particularly given the inherent class imbalance in driving datasets where "stuff" classes (road, sky, vegetation) occupy significantly larger pixel areas than "things" classes (vehicles, pedestrians, traffic signs).

As future work, we will explore other crucial factors that can further improve the quality of PLs and select most informative regions, thereby reducing the number of calls to the API, making black-box distillation from API calls more economically and computationally viable. Furthermore, we will also explore making API calls for pseudo-supervision while ensuring the privacy of sensitive local client data.