Adaptive Hierarchical Certification for Segmentation using Randomized Smoothing

Alaa Anani [1 2] **Tobias Lorenz** [1] **Bernt Schiele** [2] **Mario Fritz** [1]

Abstract

Certification for machine learning is proving that no adversarial sample can evade a model within a range under certain conditions, a necessity for safety-critical domains. Common certification methods for segmentation use a flat set of fine-grained classes, leading to high abstain rates due to model uncertainty across many classes. We propose a novel, more practical setting, which certifies pixels within a multi-level hierarchy, and adaptively relaxes the certification to a coarser level for unstable components classic methods would abstain from, effectively lowering the abstain rate whilst providing more certified semantically meaningful information. We mathematically formulate the problem setup, introduce an adaptive hierarchical certification algorithm and prove the correctness of its guarantees. Since certified accuracy does not take the loss of information into account for coarser classes, we introduce the Certified Information Gain (CIG) metric, which is proportional to the class granularity level. Our extensive experiments on the datasets Cityscapes, PASCAL-Context, ACDC and COCO-Stuff demonstrate that our adaptive algorithm achieves a higher CIG and lower abstain rate compared to the current state-of-the-art certification method. Our code can be found here: https://github.com/AlaaAnani/adaptive-certify.

1. Introduction

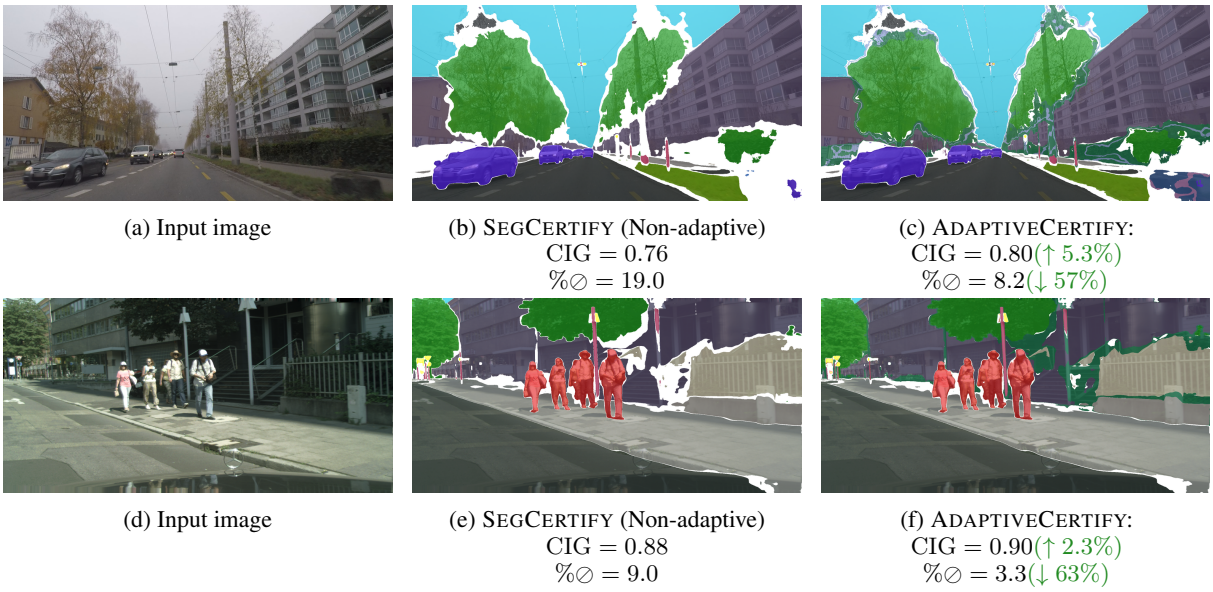
Image semantic segmentation is of paramount importance to many safety-critical applications such as autonomous driving (Kaymak & Uçar, 2019; Zhang et al., 2016), medical imaging (Kayalibay et al., 2017; Guo et al., 2019), video surveillance (Cao et al., 2020), and object detection (Gidaris & Komodakis, 2015). However, ever since deep neural networks were shown to be inherently non-robust in the face of small adversarial perturbations (Szegedy et al., 2014), the risk of using them in such applications has become evident. Moreover, an arms race between new adversarial attacks and defenses has developed, which calls for the need for provably and certifiably robust defenses. Many certification techniques have been explored in the case of classification (Li et al., 2023), with the first recent effort in semantic segmentation by Fischer et al. (2021).

Certification for segmentation is a hard task since it requires certifying many components (i.e., pixels) simultaneously. The naive approach would be to certify each component to its top class within a radius and then take the minimum as the overall certified radius of the image. This is problematic since a single unstable component could lead to a very small radius, or even abstain due to a single abstention. The state-of-the-art certification method for segmentation, SEGCERTIFY (Fischer et al., 2021), relies on randomized smoothing (Cohen et al., 2019). It mitigates the many components issue by abstaining from unstable components and conservatively certifies the rest. While an unstable component implies that the model is not confident about a single top class, it often means that it fluctuates between classes that are semantically related (a result of our analysis in Section 5 and extended in App. C.1). For example, if an unstable component fluctuates between *car* and *truck*, certifying it within a semantic hierarchy as *vehicle* would provide a more meaningful guarantee compared to abstaining.

We propose a novel hierarchical certification method for semantic segmentation, which adaptively certifies pixels within a multi-level hierarchy while preserving similar theoretical guarantees to Fischer et al. (2021). The hierarchy levels start from fine-grained labels to coarser ones that group them. Our algorithm relies on finding unstable components within an image and relaxing their label granularity to be certified within a coarser level in a semantic hierarchy. Meanwhile, stable components can still be certified within a fine-grained level. As depicted in Figure 1, our approach lowers the abstain rate while providing more certified in-

[1]CISPA Helmhotz Center for Information Security, Saarbrücken, Germany [2]Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany. Correspondence to: Alaa Anani <aanani@mpi-inf.mpg.de>, Tobias Lorenz <tobias.lorenz@cispa.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.



(a) Input image

(b) SEGCERTIFY (Non-adaptive)
CIG = 0.76
%⊘ = 19.0

(c) ADAPTIVECERTIFY:
CIG = 0.80(↑ 5.3%)
%⊘ = 8.2(↓ 57%)

(d) Input image

(e) SEGCERTIFY (Non-adaptive)
CIG = 0.88
%⊘ = 9.0

(f) ADAPTIVECERTIFY:
CIG = 0.90(↑ 2.3%)
%⊘ = 3.3(↓ 63%)

Figure 1: The certified segmentation outputs on input images (a) and (d) from SEGCERTIFY in (b) and (e), and ADAPTIVECERTIFY in (c) and (f) with their corresponding Certified Information Gain (CIG) and abstain rate %⊘. Our method provides more meaningful certified output in pixels the state-of-the-art abstains from (white pixels), with a much lower abstain rate, and higher CIG. The segmentation color palette can be found in Figure 2.

formation to the end-user compared to the state-of-the-art method.

To evaluate our method, we propose a novel evaluation paradigm that accounts for the hierarchical label space, namely the Certified Information Gain (CIG). The Certified Information Gain is proportional to the granularity of the certified label; a parent vertex (e.g., *vehicle*) has less information gain than its children (e.g., *car*, *truck*, *bus*, etc.) since it provides more general information, while leaf vertices have the most information gain (i.e., the most granular classes in the hierarchy). CIG is equivalent to certified accuracy if the defined hierarchy is flat.

Main Contributions. Our main contributions are the following: (i) We introduce adaptive hierarchical certification for image semantic segmentation by mathematically formulating the problem and its adaptation to a pre-defined class hierarchy, (ii) We propose ADAPTIVECERTIFY, the first adaptive hierarchical certification algorithm, which certifies image pixels within fine-to-coarse hierarchy levels, (iii) We employ a novel evaluation paradigm for adaptive hierarchical certification: the Certified Information Gain metric and (iv) We extensively evaluate our algorithm, showing that certifying each pixel within a multi-level hierarchy achieves a significantly lower abstain rate and higher Certified Information Gain than the current state-of-the-art certification method for segmentation. Our analysis further shows the generalization of ADAPTIVECERTIFY with respect to different noise levels and pixel types.

2. Related Works

Certification. The competition between adversarial attacks and defenses has resulted in a desire for certifiably robust approaches for verification and training (Li et al., 2023). Certification is proving that no adversarial sample can evade the model within a guaranteed region under certain conditions (Papernot et al., 2018). There are two major lines of certifiers, deterministic and probabilistic techniques.

Deterministic certification techniques such as SMT solvers (Pulina & Tacchella, 2010; 2012), Mixed-Integer Linear Programming (MILP) (Cheng et al., 2017; Dutta et al., 2018) or Extended Simplex Method (Katz et al., 2017) mostly work for small networks. To certify bigger networks, an over-approximation of the network's output corresponding to input perturbations is required (Salman et al., 2019; Gowal et al., 2019), which underestimates the robustness.

Probabilistic methods work with models with added random noise: *smoothed models*. Currently, only probabilistic certification methods are scalable for large datasets and networks (Li et al., 2023). Randomized smoothing is a probabilistic approach introduced for the classification case against l_p (Cohen et al., 2019) and non-l_p threat models (Levine & Feizi, 2020). Beyond classification, it has been used in median output certification of regression models (Chiang et al., 2020), center-smoothing (Kumar & Goldstein, 2021) to certify networks with a pseudo-metric output space, and most relevant to our work, scaled to certify semantic seg-

mentation models (Fischer et al., 2021). We expand on randomized smoothing and Fischer et al. (2021) in Section 3 to provide the necessary background for our work.

Hierarchical Classification and Semantic Segmentation. Hierarchical classification categorizes components to nodes in a class taxonomy (Silla & Freitas, 2011), which can be a tree (Wu et al., 2005) or a Directed Acyclic Graph. In a DAG, nodes can have multiple parents; trees only allow one. Classifiers vary in hierarchy depth: some require fine-grained class prediction, namely mandatory leaf-node prediction (MLNP), while others allow classification at any level, namely non-mandatory leaf-node prediction (NMLNP) (Freitas & Carvalho, 2007). One way to deal with NMLNP is to set thresholds on the posteriors to determine which hierarchy level to classify at (Ceci & Malerba, 2007).

Hierarchical classification can extend to pixel-wise segmentation. In (Li et al., 2022), a model for hierarchical semantic segmentation is introduced, using hierarchy during training. NMLNP is by no means standard in current semantic segmentation work, although from a practical perspective, it is useful for downstream tasks. Our certification for segmentation method follows an NMLNP approach: we can certify a pixel at a non-leaf node. Although we use hierarchy-related concepts from previous works, our main focus is on hierarchical certification for segmentation rather than hierarchical segmentation, using the input model as a black-box.

3. Preliminaries: Randomized Smoothing for Segmentation

In this section, we provide an overview of the essential background and notations needed to understand randomized smoothing for classification and segmentation, which we build on when we introduce our adaptive method.

Classification. The core idea behind randomized smoothing (Cohen et al., 2019) is to construct a smoothed classifier g from a base classifier f. The smoothed classifier g returns the class the base classifier f would return after adding isotropic Gaussian noise to the input x. The smooth classifier is certifiably robust to ℓ_2-perturbations within a radius R. Formally, given a classifier $f : \mathbb{R}^m \to \mathcal{Y}$ and Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, the smoothed classifier $g : \mathbb{R}^m \to \mathcal{Y}$ is defined as:

$$g(x) \coloneqq \arg\max_{a \in \mathcal{Y}} \mathbb{P}(f(x+\epsilon) = a). \tag{1}$$

Then the robustness guarantee on g is that it is robust to any perturbation $\delta \in \mathbb{R}^m$ added to x, $g(x+\delta) = g(x)$, as long as δ is ℓ_2-bounded by the certified radius: $||\delta||_2 \leq R$. To evaluate g at a given input x and compute the certification radius R, one cannot compute g exactly for black-box classifiers due to its probability component. Cohen et al. (2019) use a Monte-Carlo sampling technique to approximate g by drawing n samples from $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, evaluating $f(x+\epsilon)$ at each, and then using its outputs to estimate the top class and certification radius with a pre-set confidence of $1-\alpha$, such that $\alpha \in [0, 1)$ is the type I error probability.

Segmentation. To adapt randomized smoothing to segmentation, Fischer et al. (2021) propose a mathematical formulation for the problem and introduce the scalable SEGCERTIFY algorithm to certify any segmentation model $f : \mathbb{R}^{N \times m} \to \mathcal{Y}^N$, such that N is the number of components (i.e., pixels), and $\mathcal{Y}$ is the classes set. The direct application of randomized smoothing is done by applying the certification component-wise on the image. This is problematic since it gets affected dramatically by a single bad component by reporting a small radius or abstaining from certifying all components. SEGCERTIFY circumvents the bad components issue by introducing a strict smooth segmentation model, that abstains from a component if the top class probability is lower than a threshold $\tau \in [0, 1)$. The smooth model $g^\tau : \mathbb{R}^{N \times m} \to \hat{\mathcal{Y}}^N$ is defined as:

$$g_i^\tau(x) = \begin{cases} c_{A,i} & \text{if } \mathbb{P}_{\epsilon \sim \mathcal{N}(0,\sigma^2 I)}(f_i(x+\epsilon)) > \tau, \\ \oslash & \text{otherwise} \end{cases} \tag{2}$$

where $c_{A,i} = \arg\max_{c \in \mathcal{Y}} \mathbb{P}_{\epsilon \sim \mathcal{N}(0,\sigma^2 I)}(f(x+\epsilon) = c)$ and $\hat{\mathcal{Y}} = \mathcal{Y} \cup \{\oslash\}$ is the set of class labels combined with the abstain label. For all components where $g_i^\tau(x)$ commits to a class (does not abstain), the following theoretical guarantee holds:

Theorem 3.1 (from (Fischer et al., 2021)). *Let $\mathcal{I}_x = \{i \mid g_i^\tau \neq \oslash, i \in 1, \ldots, N\}$ be the set of certified components indices in x. Then, for a perturbation $\delta \in \mathbb{R}^{N \times m}$ with $||\delta||_2 < R \coloneqq \sigma \Phi^{-1}(\tau)$, for all $i \in \mathcal{I}_x$: $g_i^\tau(x+\delta) = g_i^\tau(x)$.*

That is, if the smoothed model g_i^τ commits to a class, then it is certified with a confidence of $1-\alpha$ to not change its output $g_i^\tau(x) = g_i^\tau(x+\delta)$ for all perturbations that are ℓ_2-bounded by the certified radius: $\forall_\delta : ||\delta||_2 \leq R$.

To estimate g^τ, Fischer et al. (2021) employ a Monte-Carlo sampling technique in SEGCERTIFY to draw n samples from $f(x+\epsilon)$ where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, while keeping track of the class frequencies per pixel. With these frequencies, p-values are computed for hypothesis testing that either result in certification or abstain. Since there are N tests performed at once, a multiple hypothesis scheme is used to bound the probability of type I error (family-wise error rate) to α.

4. Adaptive Hierarchical Certification

Abstaining from all components with a top-class probability $\leq \tau$ as previously described is a conservative requirement.

While it mitigates the bad components effect on the certification radius by abstaining from them, those components are not necessarily "bad" in principle. Bad components have fluctuating classes due to the noise during sampling, which causes their null hypothesis to be accepted, and hence, are assigned $\oslash$. While this is a sign of the lack of the model's confidence in a single top class, it often means that the fluctuating classes are semantically related (a result of our analysis in Section 5 and App. C.1). For example, if sampled classes fluctuate between *rider* and *person*, this is semantically meaningful and can be certified under a label *human* instead of abstaining. This motivates the intuition behind our hierarchical certification approach, which relaxes the sampling process to account for the existence of a hierarchy.

Challenges: To construct a certifier that adaptively groups the fluctuating components' outputs, there are three challenges to solve: (i) **Finding fluctuating components:** The question is: how do we find fluctuating or unstable components? Using the samples that are used in the statistical test would violate it since the final certificate should be drawn from i.i.d samples, (ii) **Adaptive sampling:** Assuming fluctuating components were defined, the adjustment of the sampling process to group semantically similar labels while working with a flat base model can be tricky. The challenge is to transform a model with flat, fine-grained labels into one whose output labels are part of a hierarchy while dealing with said model as a black-box, and (iii) **Evaluation:** Given a certifier that allows a component to commit to coarser classes, we need a fair comparison to other classical flat-hierarchy certification approaches (e.g., SEGCERTIFY). It is not fair to use the certified accuracy since it does not account for the information loss in croarser classes.

We construct a generalization of the smoothed model which operates on a flat-hierarchy of fine-grained classes in Eq. 2 to formulate a hierarchical version of it. To recall in the definition, a smooth model g^τ certifies a component if it commits to a top class whose probability is $> \tau$, otherwise it abstains. The construction of g^τ deals with the model f as a black-box, that is, by plugging in any different version of f, the same guarantees in Theorem 3.1 hold. We show the mathematical formulation of how we construct a hierarchical version of the smoothed model, and discuss how we overcome the challenges associated with it in this section.

4.1. Hierarchical Certification: Formulation

To define a hierarchical version of the smoothed model, we first replace the flat-hierarchy set of classes $\mathcal{Y}$ with a pre-defined class hierarchy graph $H = (\mathcal{V}, \mathcal{E})$, where the vertex set $\mathcal{V}$ contains semantic classes and the edge set $\mathcal{E}$ contains the relation on the vertices. Second, we define a hierarchical version of f, namely $f^H : \mathbb{R}^{N \times m} \to \mathcal{V}^N$, that maps the image components (pixels) with m channels ($m = 3$ for RGB) to the vertices $\mathcal{V}$. Third, we define a hierarchical smoothed model $c^{\tau,H} : \mathbb{R}^{N \times m} \to \hat{\mathcal{V}}^N$, such that $\hat{\mathcal{V}} = \mathcal{V} \cup \{\oslash\}$:

$$c^{\tau,H}{}_i(x) = \begin{cases} v_{A,i} & \text{if } \mathbb{P}_{\epsilon \sim \mathcal{N}(0,\sigma^2 I)}(f_i^H(x+\epsilon)) > \tau \\ \oslash, & \text{otherwise} \end{cases} \quad (3)$$

where $v_{A,i} = \arg\max_{v \in \mathcal{V}} \mathbb{P}_{\epsilon \sim \mathcal{N}(0,\sigma^2 I)}(f_i^H(x+\epsilon) = v)$. This certifier $c^{\tau,H}$ has three main novel components: the hierarchy graph H (Section 4.2), the hierarchical function f^H (Section 4.3) and the certification algorithm to compute $c^{\tau,H}$ (Section 4.4).

4.2. The Class Hierarchy Graph

We design the class hierarchy H used by $c^{\tau,H}$ to capture the semantic relationship amongst the classes in $\mathcal{Y}$, as illustrated in the hierarchy we build on top of the 19 classes of Cityscapes in Figure 2. The full hierarchies on all other datasets are described in App. B.1. H is a pair $(\mathcal{V}, \mathcal{E})$ representing a DAG, where $\mathcal{V}$ is the set of vertices, and $\mathcal{E}$ is

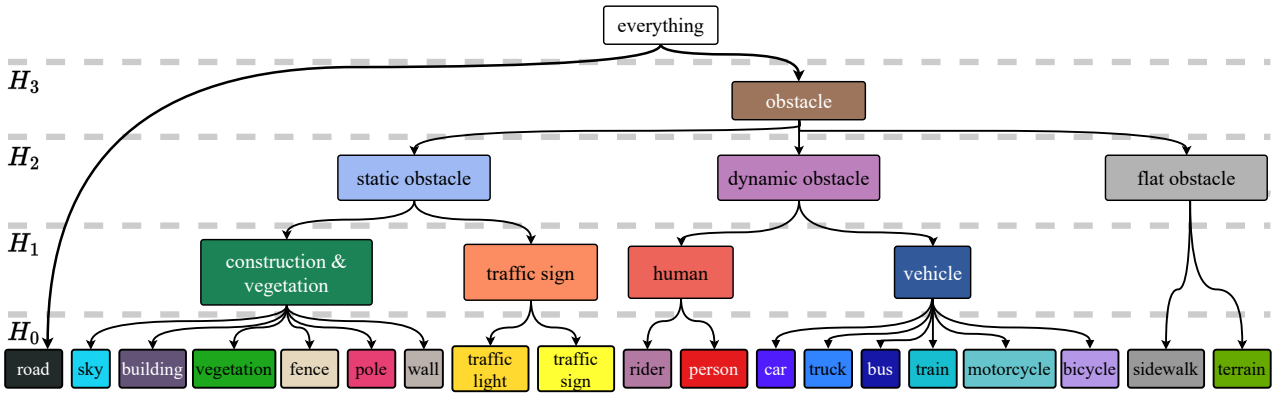


Figure 2: A DAG representing a semantic hierarchy on top of the 19 classes of Cityscapes (Cordts et al., 2016). The node colors represent the color palette used in the segmentation results. Hierarchies on all datasets are described in App. B.1.

the set of edges representing the IS-A relationship among edges. We do not allow more than one parent for each vertex. An edge $e = (u, v) \in \mathcal{E}$ is defined as a pair of vertices u and v, which entails that u is a parent of v. The root vertex of the DAG denotes the most general class, *everything*, which we do not use. The hierarchy is divided into multiple levels $H_0, \ldots, H_L$, the more fine-grained the classes are, the lower the level. A hierarchy level is a set H_l of the vertices falling within it. Leaf vertices $\mathcal{Y}$ are not parents of any other vertices. Essentially, $\mathcal{Y} = H_0$.

4.3. Fluctuating Components and Adaptive Sampling

In this part, we discuss how to solve two of the challenges concerning constructing an adaptive hierarchical certifier: defining fluctuating components without using the samples in the statistical test, and the adjustment of the sampling process to be adaptive.

We define the fluctuating components by an independent set of samples from those used in the hypothesis test. We first draw initial n_0 posterior samples per component from the segmentation head of f, defined as $f_{\text{seg}} : \mathbb{R}^{N \times m} \to [0,1]^{N \times |\mathcal{Y}|}$. We then look at the top two classes' mean posterior difference. The smaller the difference, the coarser the hierarchy level the component is assigned to. These steps are outlined in Algorithm 1 describing GETCOMPONENTLEVELS, which finds the hierarchy level index for every component.

We invoke SAMPLEPOSTERIORS to draw initial n_0 samples from $f_{\text{seg}}(x + \epsilon)$ with $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. This method retrieves n_0 posteriors per component: $Ps_1^0, \ldots, Ps_N^0$, such that Ps_i^0 is a set of n_0 posterior vectors $\in [0,1]^{|\mathcal{Y}|}$ for the i^{th} component, outlined in App. Algorithm 4. Then, for every component i, we get the mean of its n_0 posteriors P_i^0, and calculate the posterior difference ΔP_i between the top two classes, indexed by $\hat{c}_{A_i}$ and $\hat{c}_{B_i}$. We use thresholds to determine its hierarchy level index l by invoking a threshold function T_{thresh}. Given a hierarchy with L levels, the threshold function T_{thresh} is defined as:

$$T_{\text{thresh}}(\Delta P_i) = \underset{l \in \{0,\ldots,L-1\}}{\arg\min} \; t_l, \text{ s.t. } t_l < \Delta P_i \qquad (4)$$

with $t_l \in [0,1]$. T_{thresh} returns the index of the most fine-grained hierarchy level the component can be assigned to based on the pre-set thresholds $t_0 > t_1 > \ldots > t_{L-1}$.

Now that we know which level index l_i every component is mapped to, we can define f^H, which takes an image x and does a pixel-wise mapping to vertices $\hat{v}$ within every component's assigned hierarchy level H_{l_i}. Mathematically, we define the predicted label $\hat{v}_i$ for component i as:

$$f_i^H(x) = K(f_i(x), l_i) = \hat{v}_i \iff$$
$$\exists_{\hat{v}_i, u_1, \ldots, u_p, \hat{y}_i} (\{(\hat{v}_i, u_1), \ldots, (u_p, \hat{y}_i)\} \subseteq \mathcal{E}) \wedge (\hat{v}_i \in H_{l_i}) \qquad (5)$$

Algorithm 1 GETCOMPONENTLEVELS: algorithm to map components to hierarchy levels

function GETCOMPONENTLEVELS(f, x, n_0, σ)
  $Ps_1^0, \ldots, Ps_N^0 \leftarrow$ SAMPLEPOSTERIORS(f, x, n_0, σ)
  for i $\leftarrow 1, \ldots,$ N **do**
    $P_i^0 \leftarrow$ mean Ps_i^0
    $\hat{c}_{A_i}, \hat{c}_{B_i} \leftarrow$ top two class indices P_i^0
    $\Delta P_i \leftarrow P_i^0[\hat{c}_{A_i}] - P_i^0[\hat{c}_{B_i}]$
    $l_i \leftarrow T_{\text{thresh}}(\Delta P_i)$
  end for
  return $(l_1, \ldots, l_N)$, $(\hat{c}_{A_1}, \ldots, \hat{c}_{A_N})$
end function

such that there is a path from the parent vertex $\hat{v}_i$ that belongs to the hierarchy level H_{l_i} to the predicted leaf $\hat{y}_i = f_i(x)$. For example, using the hierarchy Figure 2, $K(\textit{bus}, 0) = \textit{bus}, K(\textit{bus}, 1) = \textit{vehicle}$ and $K(\textit{bus}, 2) = \textit{dynamic obstacle}$. Constructing a smoothed version of f^H, namely $c_i^{\tau,H}$, is now equivalent to the hierarchical certifier we formulated earlier in Eq. 3.

Evaluating $c^{\tau,H}$ requires a sampling scheme over f^H to get the top vertex frequencies $\text{cnts}_1, \ldots, \text{cnts}_N$ as outlined in Algorithm 2. The sampling of $f^H(x + \epsilon)$ with $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ is a form of adaptive sampling over f. For the i^{th} component with level l_i, f^H is invoked on $x + \epsilon$, which in its definition invokes $f(x + \epsilon)$ to output a flat segmentation label map $\hat{y}$, whose components $\hat{y}_i$ is mapped to its parent vertex $\hat{v}_i$ in H_{l_i} using the function K as in Eq. 5.

Algorithm 2 HSAMPLE: algorithm to adaptively sample

function HSAMPLE(f, K, $(l_1, \ldots, l_N)$, x, n, σ)
  $\text{cnts}_1, \ldots, \text{cnts}_N \leftarrow$ initialize each to a zero vector of size $|\mathcal{V}|$
  draw random noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$
  for $j \leftarrow 1, \ldots, n$ **do**
    $\hat{y} = f(x + \epsilon)$
    for $i \leftarrow 1, \ldots, N$ **do** } mathematically equivalent to calling $\equiv f^H(x + \epsilon)$
      $\hat{v}_i \leftarrow K(\hat{y}[i], l[i])$
      $\text{cnts}_i[\hat{v}_i]$ += 1
    end for
  end for
  return $\text{cnts}_1, \ldots, \text{cnts}_N$
end function

4.4. Our Algorithm: ADAPTIVECERTIFY

Putting it all together, we now introduce ADAPTIVECERTIFY 3, which overcomes the challenges of defining the fluctuating components and employing an adaptive sampling scheme to certify an input segmentation model f given a hierarchy H. Our certification algorithm approximates

the smoothed model $c^{\tau,H}$ following a similar approach by Fischer et al. (2021).

Algorithm 3 ADAPTIVECERTIFY: algorithm to hierarchically certify and predict

function ADAPTIVECERTIFY(f, K, σ, x, n, n_0, τ, α)
  $(l_1, \ldots, l_N), (\hat{c}_{A_1}, \ldots, \hat{c}_{A_N}) \leftarrow$ GETCOMPONENTLEVELS(f, x, n_0, σ)
  $\hat{v}_1, \ldots, \hat{v}_N \leftarrow$ Use $K(\hat{c}_{A_i}, l_i)$ to get parent vertices of $\hat{c}_{A_1}, \ldots, \hat{c}_{A_N}$
  $\text{cnts}_1, \ldots, \text{cnts}_N \leftarrow$ HSAMPLE($f, K, (l_1, \ldots, l_N), x, n, \sigma$)
  $pv_1, \ldots, pv_N \leftarrow$ BINPVALUE($(\hat{v}_1, \ldots, \hat{v}_N), (\text{cnts}_1, \ldots, \text{cnts}_N), \tau$)
  $\hat{v}_1, \ldots, \hat{v}_N \leftarrow$ HYPOTHESESTESTING($\alpha, \oslash, (pv_1, \ldots, pv_N), (\hat{v}_1, \ldots, \hat{v}_N)$)
  $R \leftarrow \sigma\Phi^{-1}(\tau)$
  return $\hat{v}_1, \ldots, \hat{v}_N, R$
end function

On a high level, ADAPTIVECERTIFY consists of three parts: (i) mapping components to hierarchy level indices by invoking GETCOMPONENTLEVELS, (ii) adaptively sampling to estimate the smoothed model $c^{\tau,H}$ by invoking HSAMPLE, and (iii) employing multiple hypothesis testing via HYPOTHESESTESTING (outlined in App. Algorithm 5) to either certify a component or assign $\oslash$ to it. To avoid invalidating our hypotheses test, we use the initial set of n_0 independent samples drawn in GETCOMPONENTLEVELS to both decide on the assigned component levels indices $l_1, \ldots, l_N$, as well as the top class indices $\hat{c}_{A_1}, \ldots, \hat{c}_{A_N}$. Since those classes are in $\mathcal{Y}$ as they come from the flat model f, we transform them using the mapping function K and the levels to get their corresponding parent vertices in the hierarchy: $\hat{v}_1, \ldots, \hat{v}_N$. These vertices are used to decide on the top vertex class, while the counts $\text{cnts}_1, \ldots, \text{cnts}_N$ drawn from the adaptive sampling function HSAMPLE are used in the hypothesis testing. With these counts, we perform a one-sided binomial test on every component to retrieve its p-value, assuming that the null hypothesis is that the top vertex class probability is $\leq \tau$. Then, we apply HYPOTHESESTESTING (App. Algorithm 5) to reject (certify) or accept (abstain by overwriting $\hat{v}_i$ with $\oslash$) from components while maintaining an overall type I error probability of α.

We now show the soundness of ADAPTIVECERTIFY using Theorem 3.1. That is, if ADAPTIVECERTIFY returns a class $\hat{v}_i \neq \oslash$, then with probability $1-\alpha$, the vertex class is certified within a radius of $R := \sigma\Phi^{-1}(\tau)$.

Proposition 1 (Similar to (Fischer et al., 2021)). Let $\hat{v}_1, \ldots, \hat{v}_N$ be the output of ADAPTIVECERTIFY given an input image x and $\hat{I}_x := \{i \mid \hat{v}_i \neq \oslash\}$ be the set of non-abstain indices in x. Then with probability at least $1-\alpha$, $\hat{I}_x \subseteq I_x$ such that I_x denotes the theoretical non-abstain indices previously defined in Theorem 3.1 by replacing g^τ with our smoothed model $c^{\tau,H}$. Then, $\forall i \in \hat{I}_x$, $\hat{v}_i = c_i^{\tau,H}(x) = c_i^{\tau,H}(x+\delta)$ for ℓ_2-bounded noise $||\delta||_2 \leq R$.

Proof. With probability α, a type I error would result in $i \in \hat{I}_x \setminus I_x$. However, since α is bounded by HYPOTHESESTESTING, then with probability at least $1-\alpha$, $\hat{I}_x \subseteq I_x$. □

4.5. Properties of ADAPTIVECERTIFY

The hierarchical nature of ADAPTIVECERTIFY means that instead of abstaining for unstable components, it relaxes the certificate to a coarser hierarchy level. While not always successful, this increases the chances for certification to succeed on a higher level. The abstaining can still occur on any hierarchy level, and it has two reasons: the top vertex probability is $\leq \tau$, and by definition $c^{\tau,H}$ would abstain, or it is a type II error in ADAPTIVECERTIFY.

ADAPTIVECERTIFY guarantees that the abstain rate is always less than or equal to a non-adaptive flat-hierarchy version (e.g., SEGCERTIFY). If our algorithm only uses level H_0 for all components, the abstain rate will be equal to a non-adaptive version. So, since some components are assigned to a coarser level, their p-values can only decrease, which can only decrease the abstain rate.

By adapting the thresholds in T_{thresh} and the hierarchy definition, one can influence the hierarchy levels assigned to the components. Strict thresholds or coarser hierarchies would allow most components to fall within coarse levels. This is a parameterized part of our algorithm that can be adjusted based on the application preferences, trading off the certification rate versus the Certified Information Gain. We explore the tradeoff in App. C.6.

4.6. Evaluation Paradigm: Certified Information Gain

As mentioned previously, certified accuracy does not take the loss of information into account when traversing coarser hierarchy nodes as it would be trivial to maximize certified robustness by assigning all components to the topmost level. We therefore define a new Certified Information Gain (CIG) metric that is proportional to the class granularity level. That is, a pixel gets maximum CIG if certified within the most fine-grained level H_0, and it decreases the higher the level.

Formally, given an image x with predicted certified vertices $\hat{v} = c^{\tau,H}(x)$, ground truth y, and hierarchy level map L, CIG is defined as:

$$\text{CIG}(\hat{v}, y, L) = \frac{\sum_{\hat{v}_i = K(y_i, L_i)} (\log(|\mathcal{Y}|) - \log(G(\hat{v}_i)))}{N \cdot \log(|\mathcal{Y}|)} \quad (6)$$

$|\mathcal{Y}|$ is the number of leaves (i.e., the number of the most

fine-grained classes), and $\mathrm{G}(v_i)$ returns the generality of the vertex v_i, which is defined as the number of leaf descendants of v_i (formal definition in App. Eq. 9). Assuming certification succeeds, CIG is maximized when $\hat{v}$ are all leaf vertices since $\mathrm{CIG}(\hat{v}, y, L) = \frac{\sum_{i=1}^{N}(\log(|\mathcal{Y}|)-0)}{N \cdot \log(|\mathcal{Y}|)} = 1$ as the generality of a leaf vertex $G(\hat{v}_i) = 1$. CIG results in a score between 0 and 1 and reduces to certified accuracy for non-adaptive algorithms (SEGCERTIFY).

We also consider the class-average CIG, namely cCIG, which evaluates the performance on a per-class basis. It is defined by measuring the per-class CIG for all classes in $\mathcal{Y}$ and then getting the average. Given any class $a \in \mathcal{Y}$, the per-class CIG is defined as CIG^a:

$$\mathrm{CIG}^a(\hat{v}, y, L) = \frac{\sum\limits_{y_i=a} \mathrm{CIG}_i(\hat{v}, y, L)}{|\{i \mid y_i = a\}|} \tag{7}$$

where $\mathrm{CIG}_i(\hat{v}, y, L)$ denotes the CIG of the i^{th} component. Therefore, the class-average CIG (cCIG) is expressed as:

$$c\mathrm{CIG}(\hat{v}, y, L) = \frac{1}{|\mathcal{Y}|} \sum_{a \in \mathcal{Y}} \mathrm{CIG}^a(\hat{v}, y, L) \tag{8}$$

cCIG is a score between 0 and 1, and reduces to class-average certified accuracy for non-adaptive algorithms.

5. Results

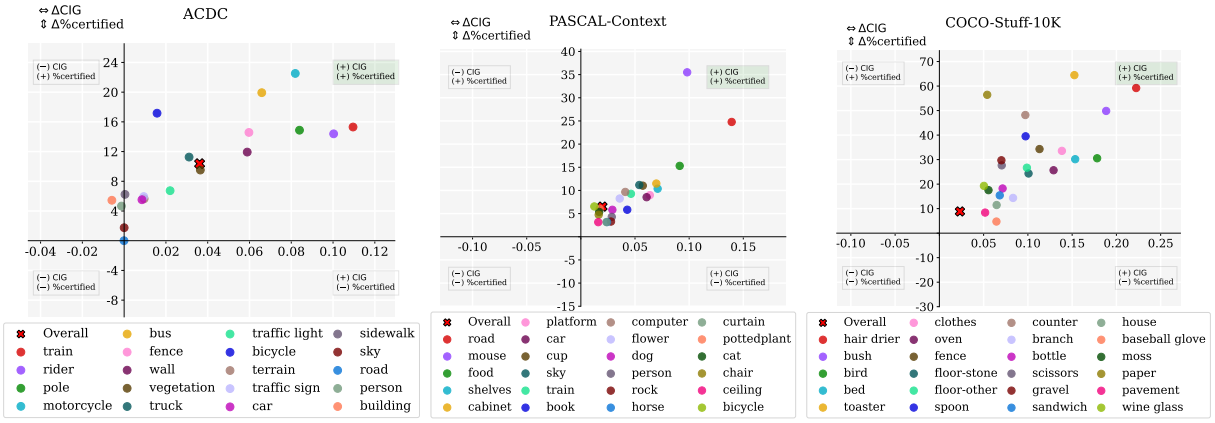
We evaluate ADAPTIVECERTIFY in a series of experiments to show its performance against the current state-of-the-art, and illustrate its hierarchical nature. We use four segmentation datasets: Cityscapes (Cordts et al., 2016), the Adverse Conditions Dataset with Correspondences (ACDC) (Sakaridis et al., 2021), PASCAL-Context (Mottaghi et al., 2014) and COCO-Stuff-10K (Caesar et al., 2018), which are described in App. B.1 alongside their hierarchy graphs.

We use HrNetV2 (Sun et al., 2019; Wang et al., 2020) as the uncertified base model, trained on Gaussian noise with $\sigma = 0.25$ as detailed in App. B.2. We use different parameters for the threshold function T_{thresh} (Eq 4) for ADAPTIVECERTIFY per dataset as described in App. B.3, which we found via a grid search that maximizes the CIG metric. The evaluation metrics include CIG, cCIG, $\%\oslash$ (percentage of abstain pixels), or its complement: $\%$certified (percentage of certified pixels), and $c\%\oslash$ (class-averaged $\%\oslash$). All details on the experimental setup can be found in App. B.

We first investigate the overall performance of ADAPTIVECERTIFY against the baseline SEGCERTIFY across noise levels σ and number of samples n in Table 1. On a high level, ADAPTIVECERTIFY consistently has a higher CIG and lower $\%\oslash$ than SEGCERTIFY. Although increasing the noise level σ degrades the performance in both algorithms, ADAPTIVECERTIFY abstains much less than SEGCERTIFY, while maintaining a higher CIG, at higher noise levels. The improvement in CIG and $\%\oslash$ is highest on the COCO-Stuff-10K dataset, at 3.4% and 35%. COCO-Stuff-10K has a large number of classes –171– best highlighting the efficacy of our hierarchical certification approach. We investigate the overall performance in more detail, including the mIoU,

				Cityscapes		ACDC		PASCAL-Context		COCO-Stuff-10K	
		σ	R	CIG ↑	$\%\oslash$ ↓	CIG ↑	$\%\oslash$ ↓	CIG ↑	$\%\oslash$ ↓	CIG ↑	$\%\oslash$ ↓
	Uncertified HrNet	-	-	0.90	–	0.61	–	0.58	–	0.65	–
$n = 100$, $\tau = 0.75$	SEGCERTIFY	0.25	0.17	0.89	7	0.67	21	0.57	20	0.58	20
		0.33	0.22	0.81	14	0.57	27	0.46	31	0.52	28
		0.50	0.34	0.41	26	0.25	26	0.15	41	0.31	45
	ADAPTIVECERTIFY	0.25	0.17	**0.90** 1.1%	**5** 28.6%	**0.68** 1.5%	**16** 23.8%	**0.58** 1.8%	**16** 20.0%	**0.60** 3.4%	**13** 35.0%
		0.33	0.22	**0.83** 2.5%	**10** 28.6%	**0.59** 3.5%	**22** 18.5%	**0.48** 4.3%	**26** 16.1%	**0.54** 3.8%	**18** 35.7%
		0.50	0.34	**0.44** 7.3%	**15** 42.3%	**0.27** 8.0%	**18** 30.8%	**0.16** 6.7%	**36** 12.2%	**0.35** 12.9%	**32** 28.9%
$n = 500$, $\tau = 0.95$	SEGCERTIFY	0.25	0.41	0.86	12	0.63	29	0.53	30	0.53	31
		0.33	0.52	0.76	22	0.51	39	0.40	46	0.45	41
		0.50	0.82	0.36	39	0.22	39	0.12	59	0.26	61
	ADAPTIVECERTIFY	0.25	0.41	**0.87** 1.2%	**9** 25.0%	**0.64** 1.6%	**25** 13.8%	**0.54** 1.9%	**26** 13.3%	**0.56** 5.7%	**25** 19.4%
		0.33	0.52	**0.77** 1.3%	**18** 18.2%	**0.53** 3.9%	**34** 12.8%	**0.41** 2.5%	**41** 10.9%	**0.48** 6.7%	**33** 19.5%
		0.50	0.82	**0.40** 11.1%	**28** 28.2%	**0.24** 9.1%	**31** 20.5%	**0.13** 8.3%	**55** 6.8%	**0.29** 11.5%	**50** 18.0%

Table 1: Certified segmentation results for 200 images from each dataset. We extend this table by including the mIoU, cCIG and $c\%\oslash$ metrics per dataset in App. Tables 4, 5, 6 and 7 under App. C.2.



(a) (b) (c)

Figure 3: The performance of AdaptiveCertify against the baseline in terms of the difference in CIG (ΔCIG) and the certification rate (Δ%certified) across the 3 datasets w.r.t their top classes. "Overall" indicates the class-average performance. Extensions of this figure to Cityscapes and all classes in PASCAL-Context and COCO-Stuff-10K datasets are in App. Figures 14, 15 and 16 under App. C.5.

cCIG and $c\%\oslash$ metrics per dataset in App. C.2.

We investigate the per-class performance of AdaptiveCertify and SegCertify in Figure 3 in terms of CIG and %certified on the datasets: ACDC, PASCAL-Context and COCO-Stuff-10K. The average overall difference in performance shows that we certify 12%, 5% and 10% more of the pixels in all 3 datasets respectively, while achieving a higher CIG by ≈ 0.04 on ACDC and ≈ 0.02 on the rest. Almost all of the classes lie in the quadrant where AdaptiveCertify outperforms SegCertify across both metrics (upper right quadrant), reaching a maximum improvement in CIG of $+0.23$ in the class *hair drier* in COCO-Stuff-10K and %certified increase of $+65$ percentage points in the class *toaser* in the same dataset. The performance remains the same (with a Δ of 0) for leaf classes with no parent vertices at coarser hierarchy levels, such as *road* in ACDC (hierarchy in Figure 2), since by definition AdaptiveCertify is reduced to SegCertify in a single-level hierarchy.

To illustrate the hierarchical nature of AdaptiveCertify, we plot the pixel distribution across hierarchy levels in Figure 4. We observe that both methods certify a comparable number of pixels at the finest level H_0. However, due the hierarchical structure of AdaptiveCertify, there is a notable advantage in certifying additional percentages (ranging from 2% to 7%) of pixels at higher hierarchy levels in the 4 datasets, where SegCertify opts to abstain. This effect is the strongest in the more challenging datasets ACDC, PASCAL-Context and COCO-Stuff-10K. Due to more fluctuating components, more pixels are assigned to coarser hierarchy levels.

Boundary pixels constitute a challenge in segmentation,

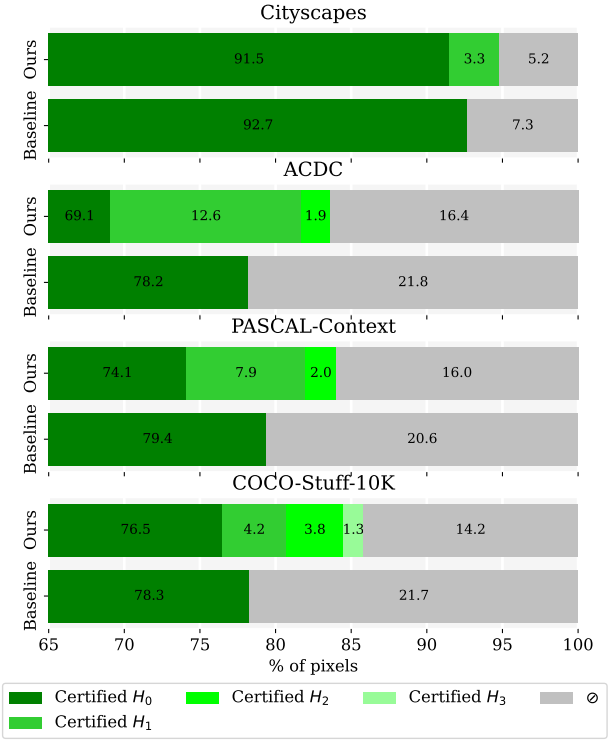


Figure 4: The performance of our algorithm against the baseline in terms of percentage of abstain and certified pixels under different hierarchy levels. SegCertify, by definition, only uses H_0.

prompting an analysis of the certification performance of AdaptiveCertify against a non-adaptive baseline (SegCertify) on them, as shown in Table 2. We explain

	Cityscapes			ACDC			PASCAL-Context			COCO-Stuff-10K		
	Baseline	Ours	%Δ	Baseline	Ours	%Δ	Baseline	Ours	%Δ	Baseline	Ours	%Δ
⊘% ↓	7.3	5.2	28.8%	21.8	16.4	24.8%	20.6	16.1	21.8%	21.7	14.2	34.6%
⊘%boundary ↓	19.3	13.7	29.0%	35.0	25.0	28.6%	26.8	22.2	17.2%	26.4	19.3	26.9%
⊘%non-boundary ↓	5.1	3.7	27.5%	20.2	15.4	23.8%	20.0	15.5	22.5%	20.5	12.9	37.1%
CIG ↑	0.89	0.90	1.12%	0.62	0.63	1.61%	0.55	0.56	1.82%	0.52	0.54	3.85%
CIG-boundary ↑	0.71	0.72	1.41%	0.43	0.45	4.65%	0.37	0.38	2.70%	0.44	0.46	4.55%
CIG-non-boundary ↑	0.92	0.93	1.09%	0.64	0.65	1.56%	0.56	0.58	3.57%	0.54	0.56	3.70%

Table 2: Mean per-pixel certification performance of ADAPTIVECERTIFY against SEGCERTIFY over the first 100 images from each dataset. ⊘%boundary$=\frac{\text{number of } \oslash \text{ and boundary pixels}}{\text{number of boundary pixels}}$ and CIG-boundary $=$ CIG of boundary pixels only, and a similar logic follows for non-boundary metrics.

the setup to isolate boundary pixels in App. C.4. Overall, ADAPTIVECERTIFY maintains a positive percentage improvement (%Δ) on both boundary and non-boundary pixels across all metrics. A higher percentage of boundary pixels is abstained from by both methods compared to the non-boundary pixels, with a maximum of 35% in the challenging ACDC dataset by the baseline. Similarly, the CIG of the boundary pixels is lower in both methods. This is attributed to the difficulty of segmenting boundary pixels as they mark label transitions in the segmentation map. However, the %Δ improvement of ADAPTIVECERTIFY over the baseline in boundary pixels is on average higher than that of non-boundary pixels, with the exception of the PASCAL-Context dataset. This shows the effectiveness of the hierarchical grouping of labels in our adaptive method, especially on pixels the model is not confident about at object boundaries. We show qualitative analysis of a visual example in App. C.4.

Abstentions occur due to model uncertainty caused by noise perturbations, indicating a lack of confidence in a single top class. We investigate the fluctuating output classes of the model by visualizing recurring class sets in unstable components (e.g., abstain pixels by SEGCERTIFY) in two datasets, shown in Figure 5. We observe these recurring sets aligning with higher-level semantic concepts; for instance, (*grass*, *mountain*, *sky*) are grouped under *nature* in PASCAL-Context hierarchy in App. Figure 6, while (*dirt*, *grass*, *gravel*, *ground-other*, *sand*) fall under *ground* in COCO-Stuff (Caesar et al., 2018). Our approach groups fluctuating classes under broader labels in a semantic hierarchy, leading to fewer abstentions.

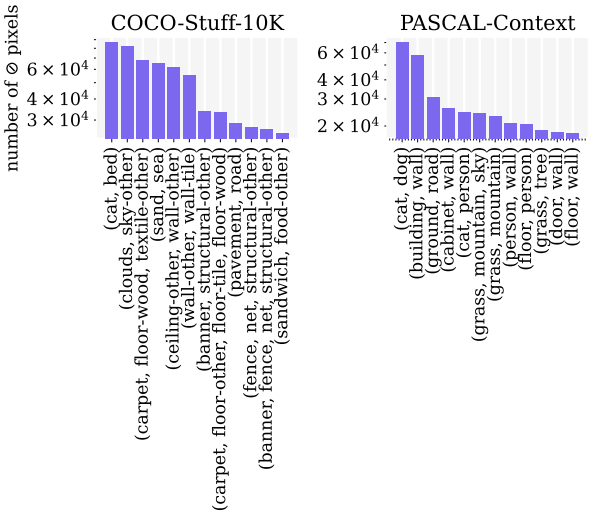


Figure 5: The frequency of the top-most sets of classes the model fluctuates between in abstain pixels by the baseline on the COCO-Stuff-10K and PASCAL-Context datasets.

6. Conclusion

In this paper, we investigate the problem of high abstain rates in common certification techniques that rely on a flat hierarchy. Based on that, we introduce adaptive hierarchical certification for semantic segmentation. We mathematically formulate the problem and propose ADAPTIVECERTIFY, that solves three main challenges: finding unstable components, adaptive sampling from multiple hierarchy levels, and evaluating the results using the Certified Information Gain metric. Instead of abstaining for unstable components, ADAPTIVECERTIFY relaxes the certification to a coarser hierarchy level. It guarantees an abstain rate less than or equal to non-adaptive versions while maintaining a higher CIG across different noise levels and number of samples. Considering that boundary pixels constitute a challenge in segmentation, ADAPTIVECERTIFY's improvement is widespread across both boundary and non-boundary pixels. The formulation of our hierarchical certification method is general and can adopt any hierarchy graph.

Impact Statement

In this paper, we introduced adaptive hierarchical certification for semantic segmentation, in which certification can be within a multi-level hierarchical label space, which lowers the abstain rate and increases the certified information gain compared to conventional methods. We adapt and relax the certification for challenging pixels, by certifying them to a coarser label in a semantic hierarchy, rather than abstaining. Our adaptive hierarchical certification method not only addresses the limitations of conventional certification methods but also presents ethical considerations and potential societal consequences in domains crucial to human well-being. The overall impact is assessed to be positive as it contributes to understanding, methodology, and mitigation of robustness issues with current AI/ML methods.

Societal impact: Image semantic segmentation is of importance to safety-critical applications, including autonomous driving, medical imaging, and video surveillance. Through hierarchical certification of segmentation models deployed in these domains, we provide more certified information gain on average, particularly in challenging images. This is particularly noteworthy as conventional certification methods often encounter significant abstain rates in such complex scenarios. For instance, in the context of autonomous driving, it is more advantageous for the system to ascertain whether a group of pixels is drivable or not, rather than refraining from certification due to the constraints of attempting to certify them within numerous fine-grained classes. Collaboration with stakeholders in autonomous driving, medical research, and surveillance technology can facilitate the integration of the idea of hierarchical certification, leading to enhanced decision support systems and contributing to the overall safety and efficiency of these applications.

Ethical considerations: Our adaptive hierarchical certification provides more certified semantically meaningful information in image semantic segmentation, which contributes to ethical decision-making processes in the aforementioned safety-critical applications that rely on semantic segmentation.

Conclusion In conclusion, our paper not only introduces a novel approach to adaptive hierarchical certification for image semantic segmentation but also emphasizes its critical role in safety-critical applications. By addressing ethical considerations and anticipating future societal consequences, our work serves as a catalyst for advancements in technology. Overall, we foresee a positive impact, as the contribution addresses short comings of AI/ML method w.r.t. robustness and reliability that will ultimately lead to safer AI/ML systems.

Acknowledgements

We very much appreciate the diligent and constructive reviews by all reviewers, and believe the additional insights gained in preparing the rebuttal, and expanding the analysis in our work, significantly strengthen our paper.

This work was partially funded by ELSA – European Lighthouse on Secure and Safe AI funded by the European Union under grant agreement No. 101070617. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.

References

Bonferroni, C. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 1936.

Caesar, H., Uijlings, J., and Ferrari, V. Coco-stuff: Thing and stuff classes in context. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Cao, X., Gao, S., Chen, L., and Wang, Y. Ship recognition method combined with image segmentation and deep learning feature extraction in video surveillance. *Multimedia Tools and Applications*, 2020.

Ceci, M. and Malerba, D. Classifying web documents in a hierarchy of categories: a comprehensive study. *Journal of Intelligent Information Systems*, 2007.

Cheng, C.-H., Nührenberg, G., and Ruess, H. Maximum resilience of artificial neural networks. In *Automated Technology for Verification and Analysis*, 2017.

Chiang, P.-y., Curry, M., Abdelkader, A., Kumar, A., Dickerson, J., and Goldstein, T. Detection as regression: Certified object detection with median smoothing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*, 2019.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Dutta, S., Jha, S., Sankaranarayanan, S., and Tiwari, A. Output range analysis for deep feedforward neural networks. In *NASA Formal Methods Symposium*, 2018.

Fischer, M., Baader, M., and Vechev, M. Scalable certified segmentation via randomized smoothing. In *International Conference on Machine Learning (ICML)*, 2021.

Freitas, A. and Carvalho, A. A tutorial on hierarchical classification with applications in bioinformatics. *Research and Trends in Data Mining Technologies and Applications*, 2007.

Gidaris, S. and Komodakis, N. Object detection via a multi-region and semantic segmentation-aware cnn model. In *International Conference on Computer Vision (ICCV)*, 2015.

Gowal, S., Dvijotham, K. D., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., and Kohli, P. Scalable verified training for provably robust image classification. In *International Conference on Computer Vision (ICCV)*, 2019.

Guo, Z., Li, X., Huang, H., Guo, N., and Li, Q. Deep learning-based image segmentation on multimodal medical imaging. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 2019.

Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J. Reluplex: An efficient smt solver for verifying deep neural networks. In *Computer Aided Verification (CAV)*, 2017.

Kayalibay, B., Jensen, G., and van der Smagt, P. Cnn-based segmentation of medical imaging data. *arXiv preprint arXiv:1701.03056*, 2017.

Kaymak, Ç. and Uçar, A. A brief survey and an application of semantic image segmentation for autonomous driving. *Handbook of Deep Learning Applications*, 2019.

Kumar, A. and Goldstein, T. Center smoothing: Certified robustness for networks with structured outputs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Levine, A. and Feizi, S. Wasserstein smoothing: Certified robustness against wasserstein adversarial attacks. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Li, L., Zhou, T., Wang, W., Li, J., and Yang, Y. Deep hierarchical semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Li, L., Xie, T., and Li, B. Sok: Certified robustness for deep neural networks. In *IEEE Symposium on Security and Privacy (SP)*, 2023.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

Mottaghi, R., Chen, X., Liu, X., Cho, N.-G., Lee, S.-W., Fidler, S., Urtasun, R., and Yuille, A. The role of context for object detection and semantic segmentation in the wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Papernot, N., McDaniel, P., Sinha, A., and Wellman, M. P. Sok: Security and privacy in machine learning. In *European Symposium on Security and Privacy (EuroS&P)*, 2018.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Pulina, L. and Tacchella, A. An abstraction-refinement approach to verification of artificial neural networks. In *Computer Aided Verification (CAV)*, 2010.

Pulina, L. and Tacchella, A. Challenging smt solvers to verify neural networks. *Ai Communications*, 2012.

Sakaridis, C., Dai, D., and Van Gool, L. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *International Conference on Computer Vision (ICCV)*, 2021.

Salman, H., Yang, G., Zhang, H., Hsieh, C.-J., and Zhang, P. A convex relaxation barrier to tight robustness verification of neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Silla, C. N. and Freitas, A. A. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 2011.

Sun, K., Xiao, B., Liu, D., and Wang, J. Deep high-resolution representation learning for human pose estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Wu, F., Zhang, J., and Honavar, V. Learning classifiers using hierarchically structured class taxonomies. In *Abstraction, Reformulation and Approximation*, 2005.

Yuan, Y., Chen, X., and Wang, J. Object-contextual representations for semantic segmentation. In *European Conference on Computer Vision (ECCV)*, 2020.

Zhang, Z., Fidler, S., and Urtasun, R. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

A. Algorithm functions

The method SAMPLEPOSTERIORS retrieves n_0 posteriors per component: $Ps_1^0, \ldots, Ps_N^0$, such that Ps_i^0 is a set of n_0 posterior vectors $\in [0,1]^{|\mathcal{Y}|}$ for the i^{th} component, as outline in Algorithm 4. The function $f_{i,\text{seg}}$ returns the posteriors Ps_i of the i^{th} component, such that f_{seg} is the segmentation head of f.

Algorithm 4 SAMPLEPOSTERIORS: algorithm to sample posteriors given noise σ

function SAMPLEPOSTERIORS(f, x, n, σ)
  $f_{\text{seg}} \leftarrow$ get segmentation head from f
  for $j \leftarrow 1, \ldots, n$ **do**
    for $i \leftarrow 1, \ldots, N$ **do**
      draw random noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$
      $Ps_i^j \leftarrow f_{i,\text{seg}}(x + \epsilon)$
    end for
  end for
  return $(Ps_1^n, \ldots, Ps_N^n)$
end function

We apply multiple hypothesis testing, similar to Fischer et al. (2021), following the Bonferroni method (Bonferroni, 1936) to reject (certify) or accept (abstain by overwriting $\hat{v}_i$ with $\oslash$) the null hypotheses of components while maintaining an overall type I error probability of α. The HYPOTHESETESTING function as described is outlined in Algorithm 5.

Algorithm 5 HYPOTHESESTESTING: algorithm to perform multiple hypotheses testing while bounding the Type I error rate by α following the Bonferroni method (Bonferroni, 1936)

function HYPOTHESESTESTING(α, $\oslash$, $(pv_1, \ldots, pv_N)$, $(\hat{v}_1, \ldots, \hat{v}_N)$)
  for $i \leftarrow 1, \ldots, N$ **do**
    if $pv_i > \frac{\alpha}{N}$ **then**
      $\hat{v}_i \leftarrow \oslash$
    end if
  end for
  return $\hat{v}_1, \ldots, \hat{v}_N$
end function

B. Experimental setup

We outline the experimental setup for our evaluation experiments that are presented in Section 5. We first discuss the datasets used and their corresponding semantic DAG hierarchies (App. B.1). Note that the inference is invoked on images with their original dimension without scaling. Next, we describe the training setup for the models used per dataset (App. B.2). Lastly, we list the values for all relevant evaluation parameters, such as the threshold function parameters (App. B.3) and the certification settings (App. B.4).

B.1. Datasets and Hierarchies

Cityscapes Cityscapes (Cordts et al., 2016) is a large-scale scene-understanding dataset of urban scene images (1024×2048 px) across 50 different cities in Germany and surrounding regions. The dataset provides pixel-level annotations for 30 classes. Our evaluation focuses on the official mode of semantic segmentation benchmarks with 19 common classes found in urban street scenes. The hierarchy DAG on top of these 19 classes is illustrated in Figure 2 and is used in all ADAPTIVECERTIFY experiments related to this dataset.

PASCAL-Context PASCAL-Context is a scene-understanding dataset with detailed pixel-wise semantic labels, an extension of the PASCAL VOC 2010 dataset encompassing over 400 classes (Mottaghi et al., 2014). We evaluate this dataset with 59 foreground classes in a setup similar to the prior work from Fischer et al. (2021). The semantic hierarchy designed for the 59 classes is illustrated in Figure 6 for use in our ADAPTIVECERTIFY experiments.

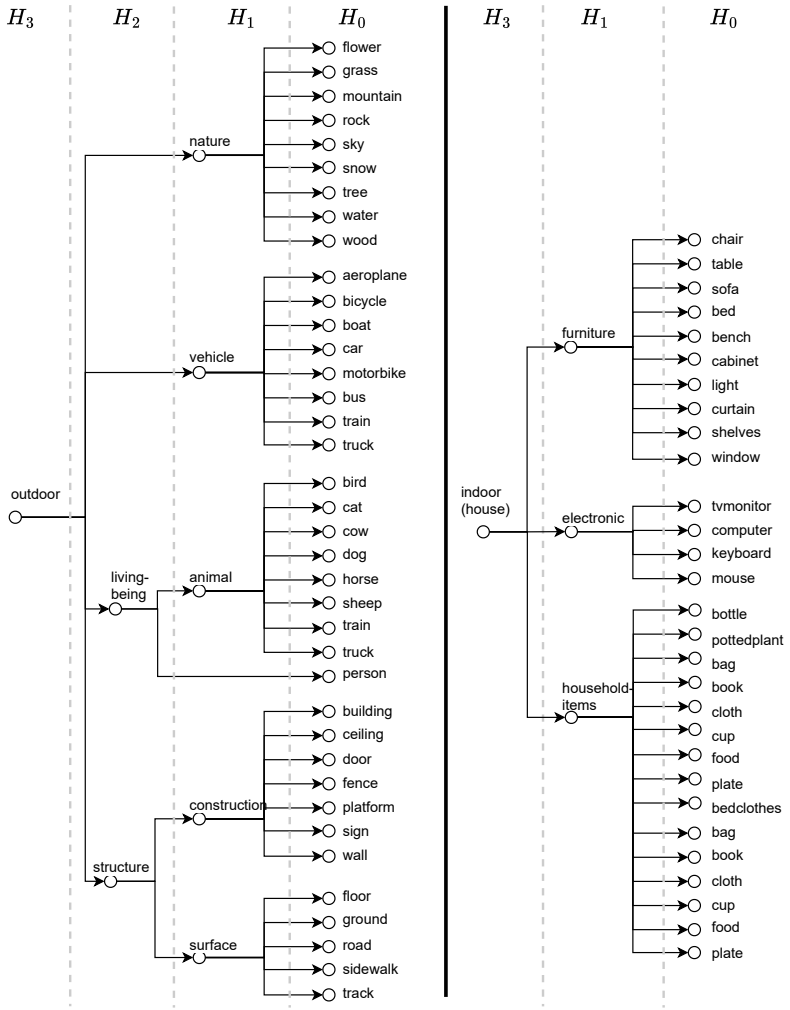


Figure 6: A DAG graph representing a semantic hierarchy on top of the 59 foreground classes of PASCAL-Context (Mottaghi et al., 2014)

Adverse Conditions Dataset with Correspondences (ACDC) ACDC is a challenging traffic scene dataset featuring images captured under four adverse visual conditions: snow, rain, fog, and nighttime (Sakaridis et al., 2021). It includes 19 classes identical to the evaluation classes in Cityscapes (H_0 in Appendix Figure 2), and we adopt the Cityscapes hierarchy DAG for this dataset.

Common Objects in COntext-Stuff (COCO-Stuff) COCO-Stuff (Caesar et al., 2018) is a scene understanding dataset, that is an extension of the COCO dataset (Lin et al., 2014), which addresses the segmentation of pixels as either thing or stuff classes. It has 172 categories (80 things, 91 stuff and 1 unlabelled). Our evaluation utilizes the common mode with 171 categories, excluding the unlabelled class. We work with the COCO-Stuff-10K v1.1 subset, consisting of 9k training and 1k validation splits. The pre-defined hierarchy for things and stuff officially provided by the dataset (Caesar et al., 2018) is used for the DAG hierarchy.

B.2. Models and Training

For the Cityscapes, ACDC and PASCAL-Context, HrNetV2 (Wang et al., 2020; Yuan et al., 2020) is used, with the HRNetV2-W48 backbone. We use the weights provided by (Fischer et al., 2021) in their official paper PyTorch (Paszke et al., 2019) implementation, which is the result of training the model on a Gaussian noise of $\sigma = 0.25$ following a similar training procedure to that of the PyTorch implementation of HrNetV2. It's important to note that for Cityscapes and ACDC, we employ the same model trained solely on Cityscapes data. This choice is intentional to evaluate the adaptive hierarchical certification method under slight domain shifts present in ACDC. The clean accuracy of the final HrNetV2 model is 90%, 61%, and 58% on the three datasets, respectively.

For the COCO-Stuff-10K dataset (Caesar et al., 2018), we used the HrNetV2 model (Wang et al., 2020; Yuan et al., 2020) with the HrNetV2-W48 Paddle Cls pre-trained backbone that follows the Object-Contextual Representations (OCR) approach, which is available in the official PyTorch implementation of the HrNetV2 paper (Paszke et al., 2019). We follow the same outlined training procedure, except by adding a Gaussian noise of $\sigma = 0.25$ in an alternating manner across the batches for the same number of 110 epochs. The final model performance on the clean split has a mean per-pixel accuracy of 62.77% and an mIoU of 0.3146. Meanwhile, on the noisy validation split, the mean per-pixel accuracy is 53.43% and the mIoU is 0.2436. We validate twice every epoch on both the clean and noisy ($\sigma = 0.25$) validation splits. During validation, we calculate the following metrics: the mean per-pixel accuracy, the mean accuracy, and the mean intersection over union (mIoU). The batch sizes used for training and validation were 12 and 1 respectively, per GPU. We validate on non-resized images with a scale of 1, and also show the certification results on them. In Figure 7, the training loss is shown in subfigure (a), with validation loss shown for both the clean and noisy validation splits in subfigures (b) and (c). Additionally, Figures 8 and 9 illustrate validation mIoU and mean per-pixel accuracy on clean and noisy validation splits during training.

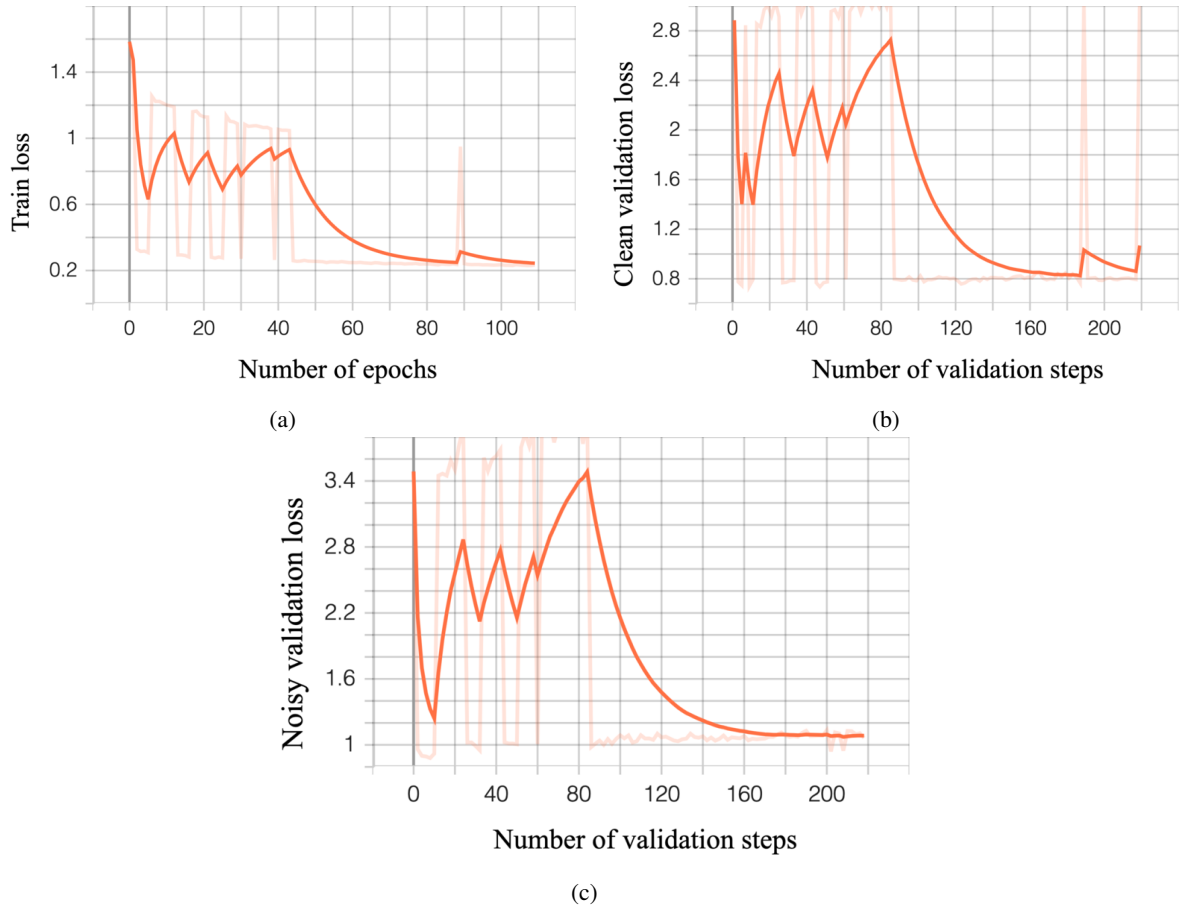
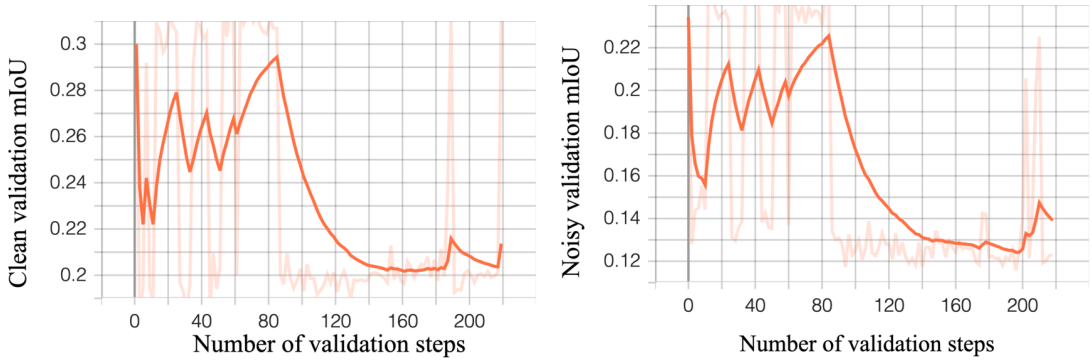
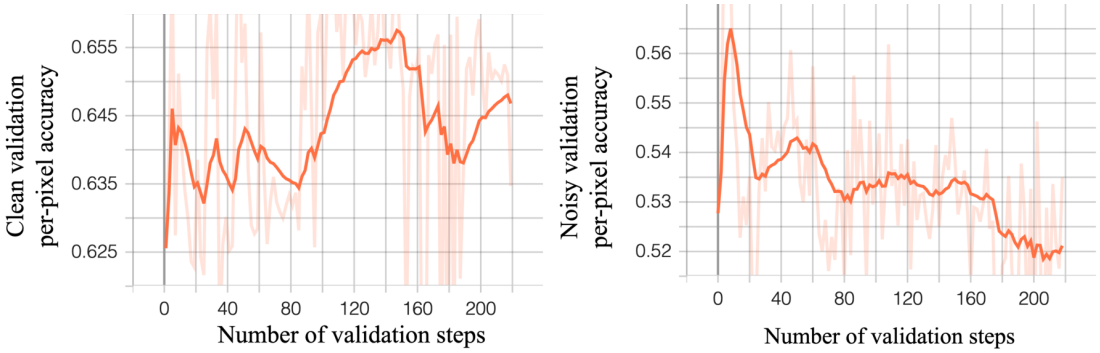


Figure 7: The train and validation loss during the training of HrNetV2 model (Wang et al., 2020; Yuan et al., 2020) on the COCO-Stuff-10K dataset (Caesar et al., 2018) on noise with $\sigma = 0.25$.



(a) (b)

Figure 8: The clean and noisy validation mIoU during the training of the HrNetV2 model on the COCO-Stuff-10K dataset.



(a) (b)

Figure 9: The clean and noisy validation mean per-pixel accuracy during the training of the HrNetV2 model on the COCO-Stuff-10K dataset.

B.3. Threshold Function

We conducted a grid search to find the best threshold function T_{thresh} parameters to use per dataset. The best parameters are chosen such that they score the maximum CIG on the first 100 samples of the test set of the dataset, fixing the rest of the certification parameters to $n_0 = 10$, $\tau = 0.75$, $\sigma = 0.25$ and $\alpha = 0.001$.

In the following Table 3, we show the threshold functions used by ADAPTIVECERTIFY for all four datasets.

Table 3: The threshold function T_{thresh} parameters used throughout our experiments in Section 5.

	Threshold function parameters
Cityscapes	(0, 0, 0.25)
PASCAL-Context	(0, 0.1, 0.4)
ACDC	(0, 0.05, 0.3)
COCO-Stuff-10K	(0, 0.3, 0.7)

We show an example of the grid search results on Cityscapes by showing the performance of different threshold functions across different number of samples n in Figure 10. We find that the best thresholds for Cityscapes that give the highest mean certified information gain compared to the rest are $(0, 0, 0.25)$ (as also mentioned in Table 3).

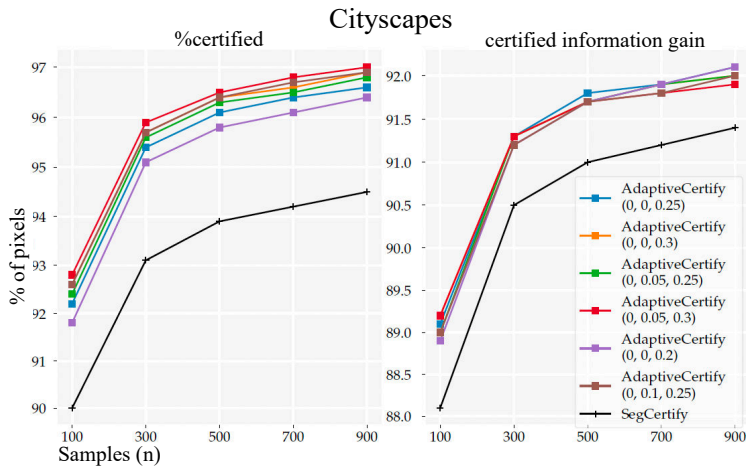


Figure 10: The performance of multiple versions of ADAPTIVECERTIFY by varying the threshold function parameters T_{thresh} against SEGCERTIFY in terms of CIG and %certified on Cityscapes. The legend is in a descending order of the mean performance. Over 63 threshold functions were used in the grid search, but we are plotting only some of them for clarity.

B.4. Certification parameters

Unless stated otherwise, all certification results use the values $\sigma = 0.25$, $\tau = 0.75$ and $\alpha = 0.001$ in both our algorithm ADAPTIVECERTIFY and the baseline SEGCERTIFY, and the metrics are the per-pixel mean over the first 100 images in each dataset.

C. Results extended

C.1. Semantic fluctuations (Extended)

The reason why randomized smoothing abstains from pixels is due to the fluctuating output of the model on perturbing such pixels with noise. It implies that the model is not confident about a single top class. Our method relies on grouping such fluctuating classes under coarser labels in higher levels in a semantic hierarchy, and thus abstaining less. In this section, we look into how those unstable components fluctuate amongst classes that are semantically related. We show the most recurring sets of classes in unstable components (e.g., abstain pixels by SEGCERTIFY) in all 4 datasets in Figure 11.

We observe that the topmost recurring sets of classes can be grouped under a higher level semantic concept. For example, we have as part of the top 10 sets:

- (*clouds*, sky-other): grouped under *sky* in the the pre-defined COCO-Stuff hierarchy (Caesar et al., 2018).
- (*dirt*, *grass*, *gravel*, *ground-other*, *sand*): grouped under *ground* in the predefined COCO-Stuff hierarchy (Caesar et al., 2018).
- (*building*, *vegetation*): grouped under *Construction and Vegetation* in the Cityscapes hierarchy in Figure 2.
- (*building*, *person*): grouped under *obstacle* in the Cityscapes hierarchy in Figure 2.
- (*floor*, *ground*): grouped under *surface* in the PASCAL-Context hierarchy we propose and use in App. Figure 6.

C.2. Overall Performance of ADAPTIVECERTIFY (Extended)

We examine the overall performance of ADAPTIVECERTIFY in comparison to the current state-of-the-art SEGCERTIFY under varying noise levels (σ) and sample sizes (n). Results across all four datasets are presented in Tables 4, 6, 5, and 7. In essence, ADAPTIVECERTIFY consistently demonstrates higher Certified Information Gain and lower abstention rates compared to SEGCERTIFY across all datasets. Despite the performance degradation observed with increased noise levels

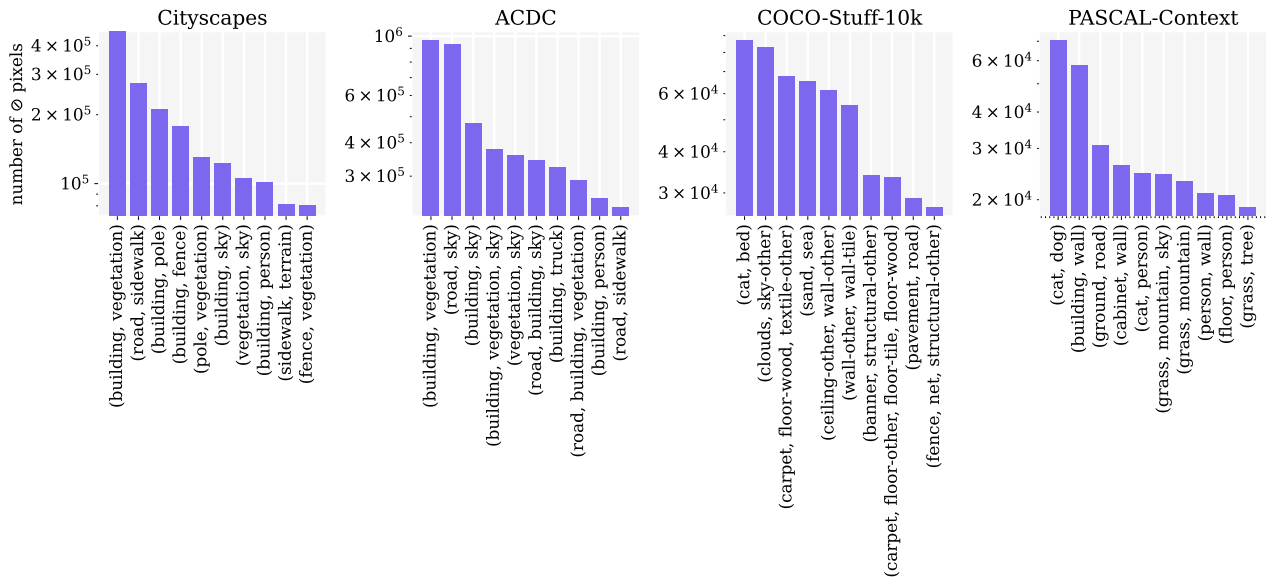


Figure 11: The frequency of the sets of classes the model fluctuates between in abstain pixels across 4 datasets: Cityscapes, ACDC, COCO-Stuff-10K and PASCAL-Context. The y-axis is a log scale. This is the result of running on 40 images per dataset.

(σ) for both algorithms, ADAPTIVECERTIFY notably exhibits reduced abstention rates while maintaining higher Certified Information Gain at even higher noise levels.

Increase in CIG Across all four datasets, there is a minimum CIG increase of 1.1%. Notably, the COCO-Stuff-10K dataset in Table 7 shows a maximum increase of 3.4% under $\sigma = 0.25$, $n = 100$, and $\tau = 0.75$. This substantial improvement in performance on the COCO-Stuff-10K dataset can be attributed to its large number of classes –171–, showcasing the hierarchy's grouping effect of many fine-grained classes under coarser labels. Similarly, the second-largest performance increase of 1.8% is observed on the PASCAL-Context dataset, also attributed to it containing the second largest class set of 59 classes.

Decrease in the abstention rate Across all four datasets, we observe a minimum decrease of 20% in the abstention rate on the PASCAL-Context dataset (Table 6), with a maximum decrease of 35% on the COCO-Stuff-10K dataset (Table 7). This significant decrease in the abstention rate can be attributed to the dataset's large number of classes, resulting in more unstable components from which the baseline tends to abstain. In contrast, our approach relaxes the certification to a coarser class, leading to a substantial improvement in performance. This improvement is further evidenced by the maximum increase in CIG, as discussed earlier.

				Cityscapes				
		σ	R	CIG ↑	cCIG ↑	%⊘ ↓	c%⊘ ↓	mIoU ↑
	Uncertified HrNet	-	-	0.90	0.47	–	–	0.39
$n = 100$, $\tau = 0.75$	SEGCERTIFY	0.25	0.17	0.89	0.65	7	21	0.37
		0.33	0.22	0.81	0.51	14	34	0.30
		0.50	0.34	0.41	0.11	26	35	0.05
	ADAPTIVECERTIFY	0.25	0.17	**0.90** 1.1%	**0.66** 1.5%	**5** 28.6%	**16** 23.8%	–
		0.33	0.22	**0.83** 2.5%	**0.53** 3.9%	**10** 28.6%	**27** 20.6%	–
		0.50	0.34	**0.44** 7.3%	**0.13** 18.2%	**15** 42.3%	**20** 42.9%	–
$n = 500$, $\tau = 0.95$	SEGCERTIFY	0.25	0.41	0.86	0.59	12	31	0.36
		0.33	0.52	0.76	0.44	22	48	0.30
		0.50	0.82	0.36	0.09	39	51	0.05
	ADAPTIVECERTIFY	0.25	0.41	**0.87** 1.2%	**0.61** 3.4%	**9** 25.0%	**26** 16.1%	–
		0.33	0.52	**0.77** 1.3%	**0.46** 4.5%	**18** 18.2%	**42** 12.5%	–
		0.50	0.82	**0.40** 11.1%	**0.11** 22.2%	**28** 28.2%	**37** 27.5%	–

Table 4: Certified segmentation results on the first 200 images in Cityscapes.

				ACDC				
		σ	R	CIG ↑	cCIG ↑	%⊘ ↓	c%⊘ ↓	mIoU ↑
	Uncertified HrNet	-	-	0.61	0.28	–	–	0.15
$n = 100$, $\tau = 0.75$	SEGCERTIFY	0.25	0.17	0.67	0.39	21	37	0.20
		0.33	0.22	0.57	0.28	27	47	0.17
		0.50	0.34	0.25	0.09	26	33	0.04
	ADAPTIVECERTIFY	0.25	0.17	**0.68** 1.5%	**0.41** 5.1%	**16** 23.8%	**29** 21.6%	–
		0.33	0.22	**0.59** 3.5%	**0.31** 10.7%	**22** 18.5%	**35** 25.5%	–
		0.50	0.34	**0.27** 8.0%	**0.11** 22.2%	**18** 30.8%	**19** 42.4%	–
$n = 500$, $\tau = 0.95$	SEGCERTIFY	0.25	0.41	0.63	0.34	29	51	0.21
		0.33	0.52	0.51	0.23	39	63	0.17
		0.50	0.82	0.22	0.08	39	49	0.04
	ADAPTIVECERTIFY	0.25	0.41	**0.64** 1.6%	**0.37** 8.8%	**25** 13.8%	**44** 13.7%	–
		0.33	0.52	**0.53** 3.9%	**0.25** 8.7%	**34** 12.8%	**52** 17.5%	–
		0.50	0.82	**0.24** 9.1%	**0.10** 25.0%	**31** 20.5%	**34** 30.6%	–

Table 5: Certified segmentation results on the first 200 images in ACDC.

				PASCAL-Context				
		σ	R	CIG ↑	cCIG ↑	%⊘ ↓	c%⊘ ↓	mIoU ↑
	Uncertified HrNet	-	-	0.58	0.22	–	–	0.15
$n = 100$, $\tau = 0.75$	SEGCERTIFY	0.25	0.17	0.57	0.27	20	27	0.17
		0.33	0.22	0.46	0.22	31	40	0.14
		0.50	0.34	0.15	0.06	41	44	0.03
	ADAPTIVECERTIFY	0.25	0.17	**0.58** 1.8%	**0.29** 7.4%	**16** 20.0%	**21** 22.2%	–
		0.33	0.22	**0.48** 4.3%	**0.25** 13.6%	**26** 16.1%	**32** 20.0%	–
		0.50	0.34	**0.16** 6.7%	**0.07** 16.7%	**36** 12.2%	**39** 11.4%	–
$n = 500$, $\tau = 0.95$	SEGCERTIFY	0.25	0.41	0.53	0.24	30	41	0.17
		0.33	0.52	0.40	0.19	46	57	0.14
		0.50	0.82	0.12	0.04	59	61	0.03
	ADAPTIVECERTIFY	0.25	0.41	**0.54** 1.9%	**0.26** 8.3%	**26** 13.3%	**35** 14.6%	–
		0.33	0.52	**0.41** 2.5%	**0.21** 10.5%	**41** 10.9%	**51** 10.5%	–
		0.50	0.82	**0.13** 8.3%	**0.05** 25.0%	**55** 6.8%	**58** 4.9%	–

Table 6: Certified segmentation results on the first 200 images in PASCAL-Context.

				COCO-Stuff-10K				
		σ	R	CIG ↑	cCIG ↑	%⊘ ↓	c%⊘ ↓	mIoU ↑
	Uncertified HrNet	-	-	0.65	0.36	–	–	0.26
$n = 100$, $\tau = 0.75$	SEGCERTIFY	0.25	0.17	0.58	0.39	20	28	0.25
		0.33	0.22	0.52	0.32	28	36	0.21
		0.50	0.34	0.31	0.15	45	53	0.10
	ADAPTIVECERTIFY	0.25	0.17	**0.60** 3.4%	**0.41** 5.1%	**13** 35.0%	**20** 28.6%	–
		0.33	0.22	**0.54** 3.8%	**0.34** 6.3%	**18** 35.7%	**26** 27.8%	–
		0.50	0.34	**0.35** 12.9%	**0.18** 20.0%	**32** 28.9%	**40** 24.5%	–
$n = 500$, $\tau = 0.95$	SEGCERTIFY	0.25	0.41	0.53	0.34	31	42	0.24
		0.33	0.52	0.45	0.26	41	53	0.20
		0.50	0.82	0.26	0.11	61	72	0.09
	ADAPTIVECERTIFY	0.25	0.41	**0.56** 5.7%	**0.36** 5.9%	**25** 19.4%	**35** 16.7%	–
		0.33	0.52	**0.48** 6.7%	**0.29** 11.5%	**33** 19.5%	**44** 17.0%	–
		0.50	0.82	**0.29** 11.5%	**0.13** 18.2%	**50** 18.0%	**62** 13.9%	–

Table 7: Certified segmentation results on the first 200 images in COCO-Stuff-10K.

C.3. Visual results

In prior sections, we examined the quantitative performance of adaptive hierarchical certification compared to the non-adaptive baseline using the CIG and abstention rate $\%\oslash$ metrics. In this section, we look more into how the certified segmentation output looks like qualitatively by both methods. We sample images from both datasets Cityscapes and ACDC to visually evaluate the performance of ADAPTIVECERTIFY (ours) against SEGCERTIFY. In Figure 12, we show selected examples that resemble a significant improvement by ADAPTIVECERITIFY against the baseline in terms of the certified information gain (CIG) and abstain rate ($\%\oslash$).

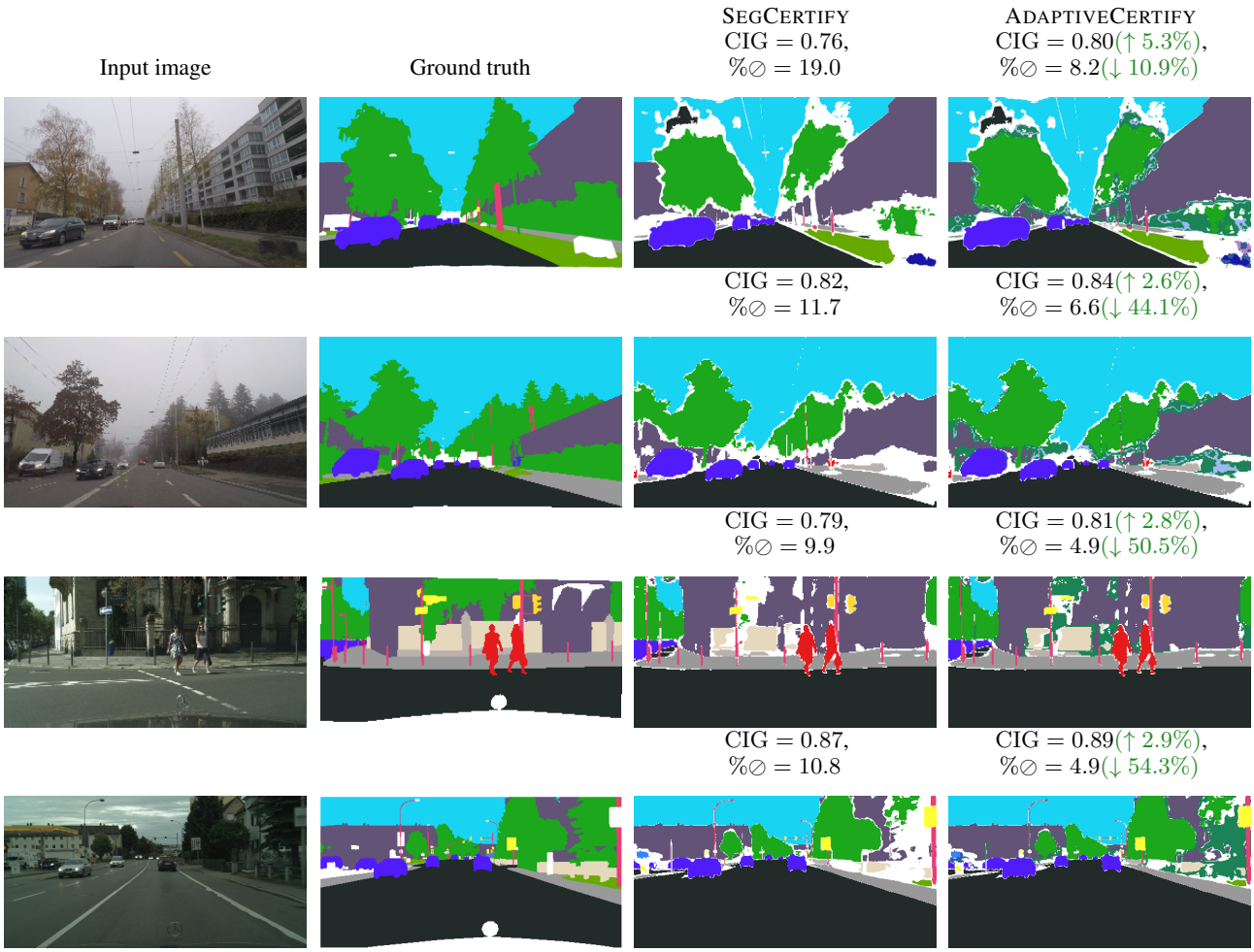
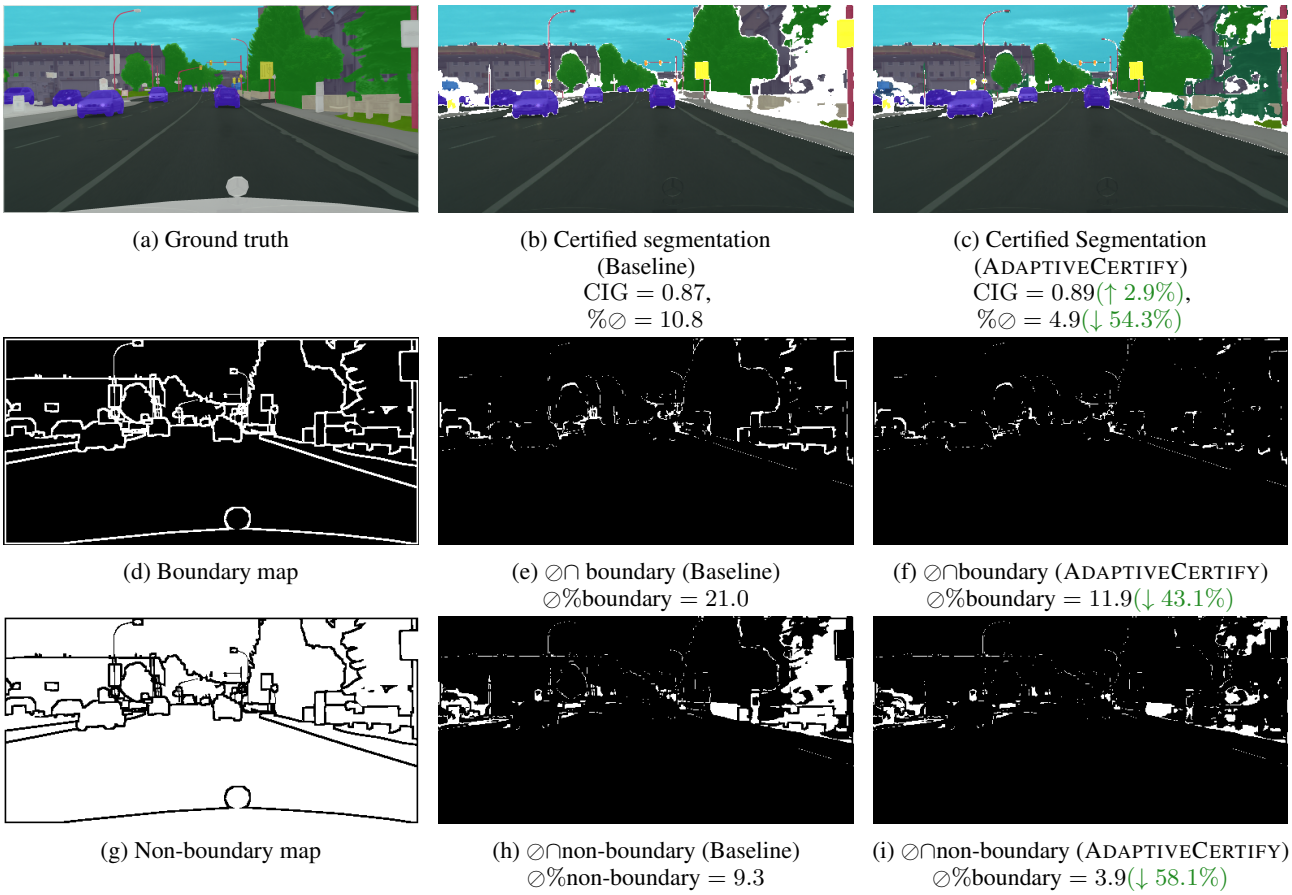


Figure 12: Selected examples showcasing the performance of ADAPTIVECERTIFY against SEGCERTIFY.

C.4. Boundary pixels analysis (Extended)

In prior sections, we demonstrated that ADAPTIVECERTIFY enhances the certification performance, quantifiable by an increase in CIG, cCIG and reduction in the abstention rate $\oslash\%$ and class-averaged abstention rate $c\oslash\%$, relative to the baseline method. Given that initial visual observations from Section C.3 suggest that the abstention predominantly occurs at object boundaries, a detailed examination was necessary. We segmented the images into boundary and non-boundary pixels to systematically analyze the abstention behavior on these pixels by both ADAPTIVECERTIFY and the baseline. This analysis aims to elucidate the specific areas where the baseline abstains, as well as show the performance of ADAPTIVECERTIFY across both pixel types.

To conduct this analysis, we need to differentiate between boundary and non-boundary pixels given a ground truth segmentation map. The way we isolate boundary pixels is by first applying 2D convolution over the ground truth segmentation map with a kernel that isolates central pixels within a defined area (controlled by the margin parameter we set to 10), followed by a dilation process that expands these central values over the surrounding pixels. The function then compares the isolated central values to the dilated values: where these differ, boundary pixels are identified, highlighting transitions between different segmentation labels. The non-boundary map is intuitively the complement of the boundary map generated via this process. An example of a boundary map is shown in Figure 13 (d) which is extracted from the ground truth segmentation map in (a).



(a) Ground truth

(b) Certified segmentation (Baseline) CIG $= 0.87$, $\%\oslash = 10.8$

(c) Certified Segmentation (ADAPTIVECERTIFY) CIG $= 0.89(\uparrow 2.9\%)$, $\%\oslash = 4.9(\downarrow 54.3\%)$

(d) Boundary map

(e) $\oslash\cap$ boundary (Baseline) $\oslash\%$boundary $= 21.0$

(f) $\oslash\cap$boundary (ADAPTIVECERTIFY) $\oslash\%$boundary $= 11.9(\downarrow 43.1\%)$

(g) Non-boundary map

(h) $\oslash\cap$non-boundary (Baseline) $\oslash\%$non-boundary $= 9.3$

(i) $\oslash\cap$non-boundary (ADAPTIVECERTIFY) $\oslash\%$boundary $= 3.9(\downarrow 58.1\%)$

Figure 13: A visual example showing how abstain pixels by the baseline (SEGCERTIFY and our method ADAPTIVECERTIFY intersect with boundary and non-boundary pixels. The first row shows the ground truth segmentation map (a) and the certified segmentation outputs by the baseline and ours in (b) and (c). The second row shows the boundary map (d) extracted from the groud truth map in (a), and the map representing the intersection between baseline and our abstain $\oslash$ pixels and boundary pixels in (e) and (f). The third row follows the same second row logic, except for non-boundary pixels.

Qualitative analysis In Figure 13, we examine the distribution of abstained pixels relative to boundary and non-boundary regions in a visual example from the Cityscapes dataset. The ground truth map displayed in (a) provides the reference for actual image boundaries, aiding in the assessment of certification outputs by SEGCERTIFY (baseline) and ADAPTIVECERTIFY (our method), shown in (a) and (b) respectively. A significant reduction in boundary pixel abstention by our method is evident in (f). Additionally, the third row features abstention analysis for non-boundary pixels, where ADAPTIVECERTIFY also demonstrates a substantial decrease in the abstention of non-boundary pixels compared to the baseline.

C.5. Per-class Performance (Extended)

An extension of Figure 3 for all classes in Cityscapes, PASCAL-Context and and COCO-Stuff-10K datasets is in Figures 14, 15 and 16, showing the per-class performance of our adaptive hierarchical method ADAPTIVECERTIFY against the baseline SEGCERTIFY.

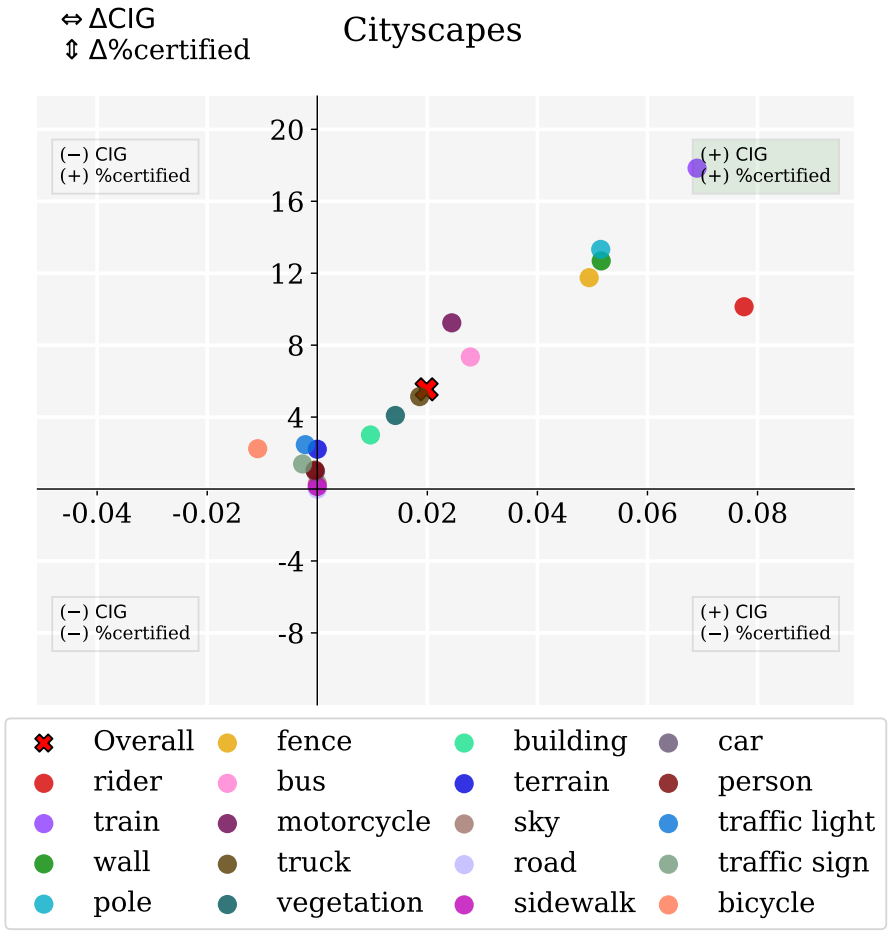


Figure 14: The performance of ADAPTIVECERTIFY against the baseline in terms of the difference in CIG (ΔCIG) and the certification rate (Δ%certified) on all classes in the Cityscapes dataset. "Overall" indicates the class-average performance. The top right quadrant indicates that ADAPTIVECERTIFY outperforms the baseline in both metrics. The results are averaged over the first 100 images in the dataset.

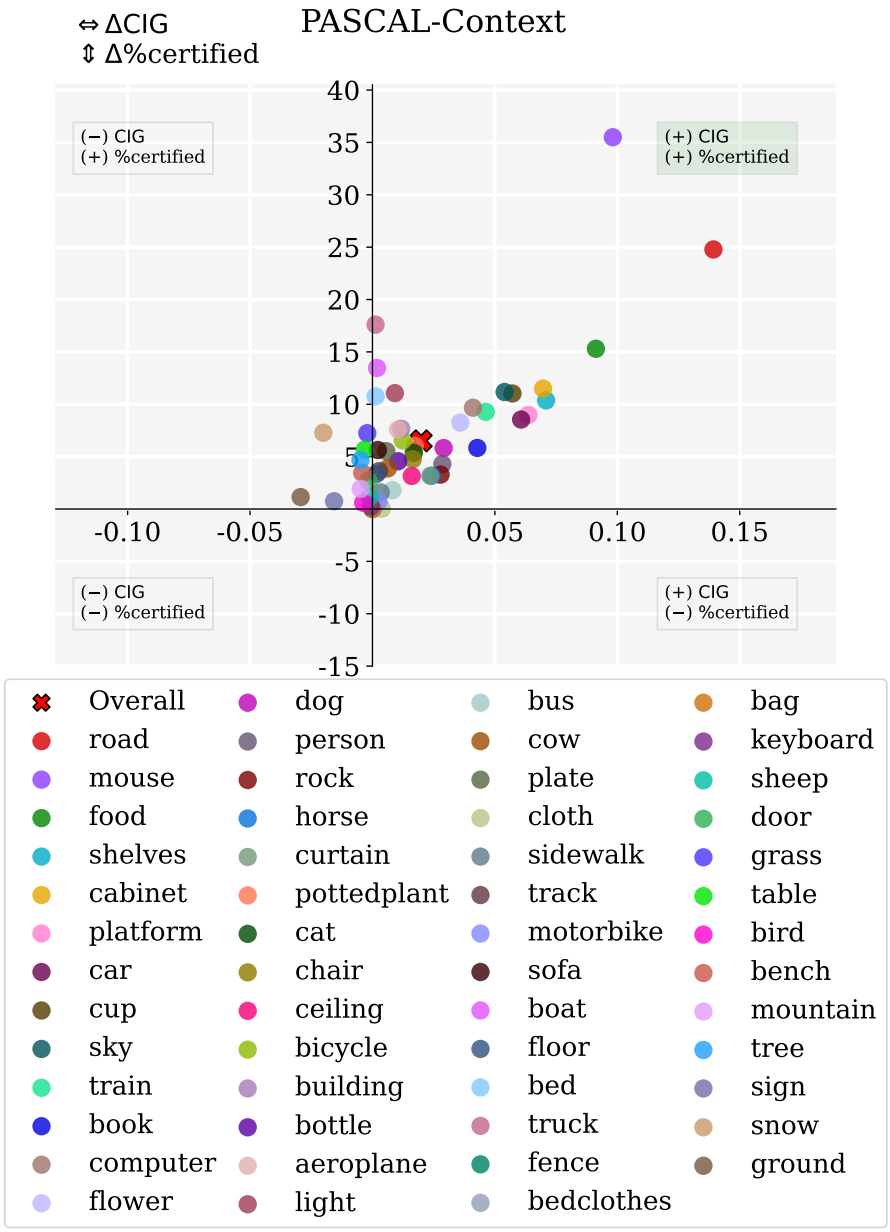


Figure 15: The performance of AdaptiveCertify against the baseline in terms of the difference in CIG (ΔCIG) and the certification rate (Δ%certified) on all classes in the PASCAL-Context dataset. "Overall" indicates the class-average performance. The top right quadrant indicates that AdaptiveCertify outperforms the baseline in both metrics. The results are averaged over the first 100 images in the dataset.

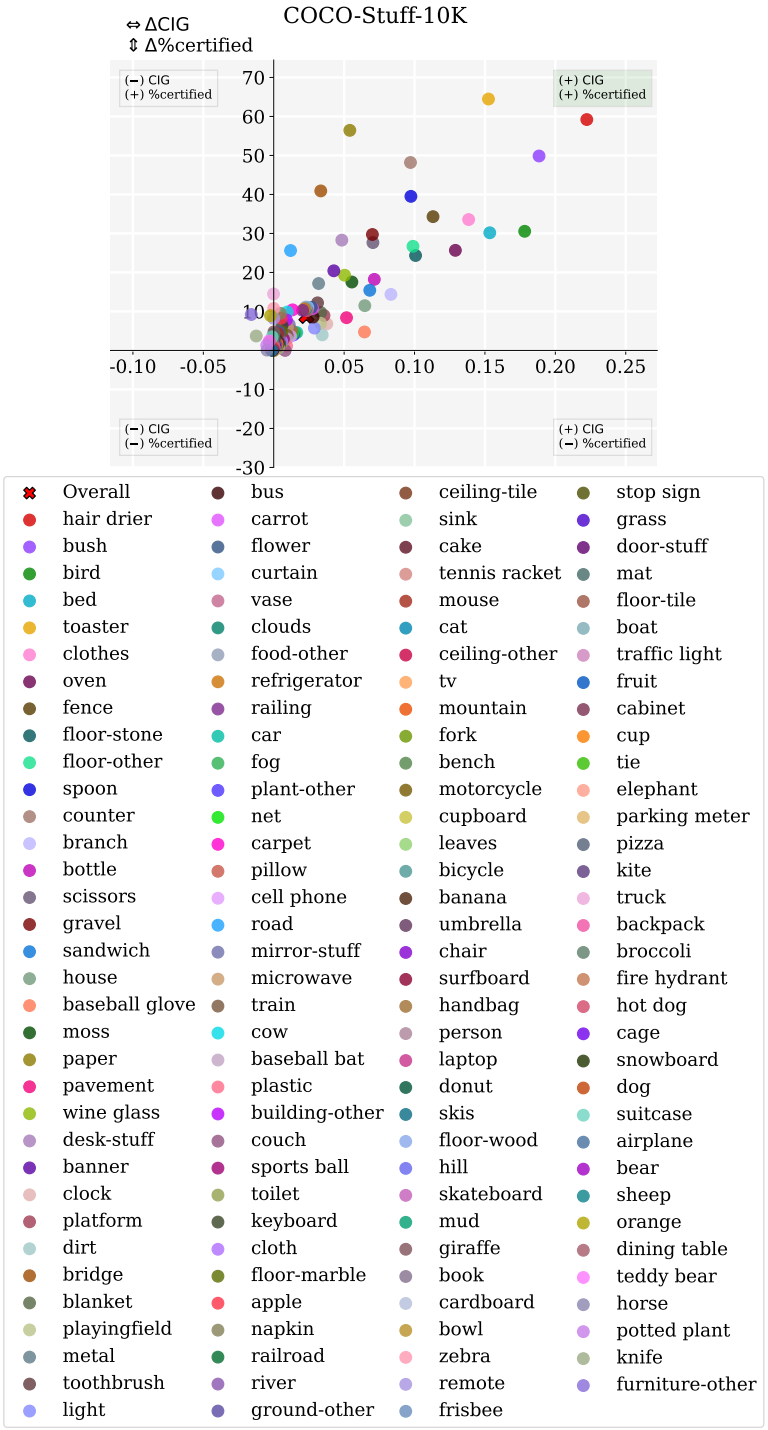
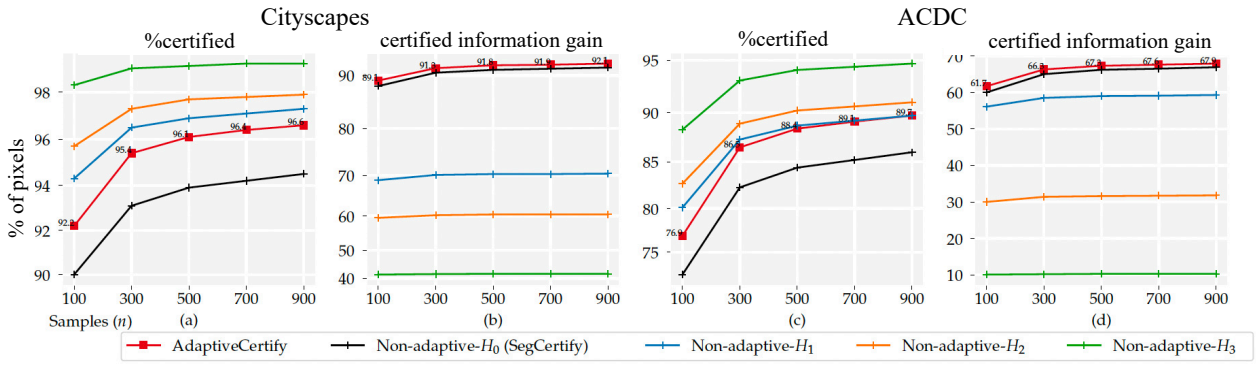


Figure 16: The performance of AdaptiveCertify against the baseline in terms of the difference in CIG (ΔCIG) and the certification rate (Δ%certified) on all classes in the COCO-Stuff-10K dataset. "Overall" indicates the class-average performance. The top right quadrant indicates that AdaptiveCertify outperforms the baseline in both metrics. The results are averaged over the first 100 images in the dataset.

C.6. CIG and Abstention Rate Tradeoff

The core idea behind adaptive hierarchical certification is to group the fluctuating classes in unstable components under a higher concept that contains them within a pre-defined semantic hierarchy. To maximize the certification rate (and minimize the abstention rate), it would be trivial, however, to group all classes under a single vertex at the most coarse level in a semantic hierarchy. This way, the certification rate will be maximized, at the cost of also having a low Certified Information Gain. On the other hand, sticking to the most fine-grained level in the hierarchy H_0 imposes a conservative requirement that causes low certification rates, as seen in SEGCERTIFY. Motivated by these two extremes—either sampling from the most coarse or fine-grained levels in the hierarchy—we look more into how adaptive hierarchical certification performs against such non-adaptive baselines.

To do so, we investigate the performance of additional baselines on Cityscapes and ACDC, where each of them samples from a different level of the hierarchy, starting from the most fine-grained H_0 (Non-adaptive-H_0 is SEGCERTIFY) up to the most coarse H_3, namely Non-adaptive-H_3, in Figure 17. We particularly want to investigate the relationship between the Certified Information Gain (CIG) and certification rate (%certified).



(a)

Figure 17: %certified (mean per-pixel certification rate) and Certified Information Gain versus the number of samples (n) on Cityscapes and ACDC.

In Figure 17, generally, for the non-adaptive baselines, the higher the level is, the higher the certification rate and lower the Certified Information Gain, and vice versa. We notice that the certification rate is the lowest in Non-adaptive-H_0 across different numbers of samples (n). Meanwhile, it is the highest for Non-adaptive-H_3, since H_3 is the most coarse level in the Cityscapes and ACDC hierarchy (refer to Figure 2 for the hierarchy). The higher the hierarchy level, the less information is retained due to the grouping of more and more classes. On the other hand, more pixels can be certified, as the number of fluctuating components decreases. By comparing the performance of the adaptive method ADAPTIVECERTIFY and the other non-adaptive baselines, we notice two things: ADAPTIVECERTIFY maintains the highest CIG across different n and σ, while it has a higher certification rate compared to Non-adaptive-H_0, but lower than the rest with H_i such that $i > 0$. Our hierarchical approach combines the best of both worlds, retaining the most amount of information from lower hierarchy levels where possible and falling back to higher hierarchy levels to avoid abstaining.

C.7. Definition: Generality of a Vertex

The generality of the vertex v_i denoted by $\mathrm{G}(v_i)$ is defined as the number of leaf vertices that are reachable by v_i, in other words, the number of leaf descendants of v_i. G is formally described as

$$G(v_i) = |\{y_j \mid \left(\exists_{v_i, v_{i-1} \ldots, v_1, y_j} \{(v_i, v_{i-1}), \ldots, (v_1, y_j)\} \subseteq \mathcal{E}\right) \wedge (y_j \in H_0)\}| \tag{9}$$

where it is the carnality of the set of leaf vertices $y_j \in H_0$ such that there is a path from the vertex v_i to y_j.