University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Matej Grcić

# SYNTHETIC NEGATIVES AND UNNORMALIZED LIKELIHOOD FOR OPEN-SET SEMANTIC SEGMENTATION OF IMAGES

DOCTORAL THESIS

Zagreb, 2024.

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Matej Grcić

# SYNTHETIC NEGATIVES AND UNNORMALIZED LIKELIHOOD FOR OPEN-SET SEMANTIC SEGMENTATION OF IMAGES

DOCTORAL THESIS

Supervisor: Professor Siniša Šegvić, PhD

Zagreb, 2024.

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Matej Grcić

# SINTETIČKI NEGATIVNI PODACI I NENORMALIZIRANA IZGLEDNOST ZA SEMANTIČKU SEGMENTACIJU SLIKA NAD OTVORENIM SKUPOM RAZREDA

DOKTORSKI RAD

Mentor: Prof. dr. sc. Siniša Šegvić

Zagreb, 2024.

Doktorski rad izrađen je na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva, na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave.

Mentor: prof. dr. sc. Siniša Šegvić

Doktorski rad ima: 84 stranice

Doktorski rad br.: _____

# O mentoru

Siniša Šegvić doktorirao je u području umjetne inteligencije i računalnog vida na zagrebačkom FER-u u 2004. godini. Bio je postdoktorski istraživač na institutu IRISA u Rennesu (2005-2006) te na TU Graz (2006-2007). Nakon toga vraća se na FER gdje predaje u području računarske znanosti i istražuje u području računalnog vida. Sudjelovao je u uvođenju diplomskih kolegija Oblikovni obrasci u programiranju, Duboko učenje te Trodimenzionalni računalni vid. Također, sudjelovao je i u rekonstrukciji kolegija Računalni vid. Konačno, sudjelovao je i u uvođenju doktorskih kolegija Analiza dinamičkih scena te Modeli za reprezentaciju slike i videa. Mentorirao je sedam obranjenih doktorata te nekoliko stotina diplomskih i završnih radova u području računalnog vida i umjetne inteligencije.

Njegovi istraživački i profesionalni interesi uključuju računalni vid, strojno učenje, razumijevanje scena, analizu satelitskih snimaka te obrane od napada putem trovanja podataka. Objavio je radove na vrhunskim konferencijama (CVPR, ECCV, NeurIPS te AAAI) te vrhunskim časopisima iz računalnog vida i umjetne inteligencije (IEEE TPAMI, IJCV, IEEE TNNLS, Patt Recog i IEEE TITS). Recenzent je u vrhunskim konferencijama i znanstvenim časopisima. Njegova istraživačka grupa sastoji se od dva postdoktoranda i šest doktoranada koje financiraju nacionalni projekti, evropski projekti i privatne tvrtke. Zajedno su postigli zapažene rezultate na više natjecanja u računalnom vidu (ACDC, WildDash, Robust vision challenge, Cityscapes, Fishyscapes i SegmentMeIfYouCan).

Vodio je tri istraživačka projekta Hrvatske zaklade za znanost (ADEPT, MultiCLOD, MASTIF), jedan projekt iz programa NPOO (VoNoMobil) te industrijska istraživanja koja su financirali Google, P3M, Rimac automobili, RoMB, MicroBlink te Promet i prostor. Sudjelovao je u istraživačkom centru izvrsnosti DataCross, projektima iz programa EDF i ERDF (EICACS, A-UNIT, SafeTram) kao i na jednom projektu iz programa FP7 (ACROSS). Sudjelovao je u industrijskom razvoju kao tehnički konzultant. Vodio je i dva bilateralna istraživačka projekta u suradnji s istraživačima iz Austrije i Njemačke te je organizirao nekoliko bilateralnih i jednu međunarodnu istraživačku radionicu.

Siniša Šegvić govori engleski i talijanski jezik te ima osnovne komunikacijske vještine na francuskom jeziku. Bio je na roditeljskom dopustu od šest mjeseci. Oženjen je i ima troje djece.

# About the Supervisor

Siniša Šegvić has received a PhD degree in computer vision and artificial intelligence at UniZg-FER. He was a postdoc researcher at IRISA Rennes (2005-2006) and at TU Graz (2006-2007). Subsequently, he returns to UniZg-FER where he lectures in computer science and performs research in computer vision. He has participated in the introduction of graduate courses Design patterns, Deep learning and Three-dimensional computer vision. He has also participated in the reconstruction of the master course Computer Vision and the introduction of doctoral courses Analysis of dynamic scenes and Models for representing images and video. He mentored seven completed doctoral theses and several hundreds of master and bachelor theses in computer vision and artificial intelligence.

His research and professional interests include computer vision, machine learning, scene understanding, recognition of satellite images, and defense from data poisoning attacks. He has published at top conferences (CVPR, ECCV, NeurIPS and AAAI) and scientific journals in computer vision and artificial intelligence (IEEE TPAMI, IJCV, IEEE TNNLS, Patt Recog). He has been a reviewer at top conferences and scientific journals. His research group consists of several postdoctoral and doctoral students that are funded by national projects, European projects and private companies. Together they achieved remarkable results to several competitions in computer vision (ACDC, WildDash, Robust vision challenge, Cityscapes, Fishyscapes and SegmentMeIfYouCan).

He has led three research projects funded by Croatian Science Foundation (ADEPT, MultiCLOD, MASTIF), one project from the Croatian RRP programme (VoNoMobil) and several industrial projects funded by Google AI for global goals, P3M, RoMB technology, Rimac Automobiles, MicroBlink and Promet i prostor. He participated in the Center of research excellence DataCross, several projects from the EDF and ERDF programmes (EICACS, A-UNIT, SafeTram) as well as one project from the FP7 programme (ACROSS). He participated in industrial development as a technical consultant. He also led two bilateral research projects in cooperation with researchers from Austria and Germany and organized several bilateral workshops and one international research workshop.

Siniša Šegvić speaks english and italian very well, and has basic communication skills in french. He had a six month career break for paternal leave. He is married and has three children.

# Zahvala

Iskreno zahvaljujem profesoru Siniši Šegviću na savjetima, strpljivosti i trudu uloženom u našu suradnju. Hvala svim kolegicama i kolegama iz istraživačke grupe profesora Šegvića na zanimljivim raspravama i ugodnoj radnoj atmosferi. Zahvaljujem profesorici Mariji Brbić i kolegama iz MLBio laboratorija na tri uzbudljiva semestra koja smo proveli zajedno na EPFL-u. Najljepše hvala majci Renati, ocu Mati, bratu Ivanu i sestri Anđeli, kao i široj obitelji i prijateljima, na bezuvjetnoj podršci i ljubavi tijekom čitavog školovanja. Konačno, zahvaljujem Jasni Galić na potpori i razumijevanju za sve moje izlete po svijetu. U najtežim trenucima nadu i snagu mi je pružao zagovor svetoga Josipa i Marije Pomoćnice.

# Abstract

Open-set segmentation considers dense recognition models that effectively handle semantic anomalies in visual input. Prominent previous approaches address this challenge by augmenting closed-set classification with dense anomaly detection. However, the existing anomaly detectors rely either on generative modelling of regular data, or discrimination with respect to negative training data. These two approaches optimize different objectives and therefore exhibit different failure modes. Consequently, this thesis proposes a novel hybrid anomaly score that fuses generative and discriminative cues. The proposed anomaly score can be incorporated into any pre-trained softmax-activated closed-set segmentation model by introducing dense estimates of the dataset posterior and unnormalized joint probability of inputs and labels. Our formulation of the joint probability preserves translational equivariance and eschews estimation of normalization constant by minimizing the density of negative training crops. Moreover, we show that real negative crops can be effectively substituted with synthetic samples from a jointly trained generative model that maximizes the likelihood of inlier crops while favouring samples with uniform discriminative prediction. Finally, we propose a novel generative architecture that increases modeling capacity by extending the coupling flows with stochastic skip connections. Our generative architecture achieves exceptional results in density estimation, and proves as the most suitable source of synthetic negatives for dense anomaly detection. Experimental results indicate that the resulting open-set segmentation models consistently outperform existing methods across benchmarks for dense anomaly detection and open-set segmentation, while incurring negligible computational overhead.

**Keywords**: semantic segmentation, open-set segmentation, open-set recognition, anomaly detection, out-of-distribution detection, synthetic negative data, generative models, normalizing flows

# Sintetički negativni podaci i nenormalizirana izglednost za semantičku segmentaciju slika nad otvorenim skupom razreda

Semantička segmentacija nad otvorenim skupom oznaka razmatra modele za gusto raspoznavanje koji učinkovito obrađuju semantičke anomalije u vizualnim ulazima. Istaknuti prethodni pristupi rješavaju ovaj izazov proširivanjem klasifikacije s gustom detekcijom anomalija. Međutim, postojeći detektori anomalija oslanjaju se ili na generativno modeliranje regularnih podataka, ili na diskriminaciju u odnosu na negativne podatke za učenje. Ova dva pristupa optimiraju različite gubitke te stoga čine različite pogreške. Stoga, ova teza predlaže novu formulaciju hibridnog detektora anomalija koji spaja generativne i diskriminativne signale. Predloženi detektor anomalija može se integrirati u bilo koji segmentacijski model sa softmax aktivacijama prethodno treniran na zatvorenom skupu uvođenjem gustih procjena posteriora skupa podataka i nenormalizirane združene vjerojatnosti ulaza i oznaka. Naša formulacija zružene vjerojatnosti čuva translacijsku ekvivarijantnost i izbjegava procjenu normalizacijske konstante minimiziranjem izglednosti negativnih isječaka. Štoviše, pokazujemo da se stvarni negativni isječci mogu učinkovito zamijeniti sintetičkim uzorcima iz zajednički učenog generativnog modela koji maksimizira vjerojatnost unutardistribucijskih isječaka, dok istovremeno favorizira uzorke s ujednačenim diskriminativnim predikcijama. Konačno, predlažemo novu generativnu arhitekturu koja povećava generativni kapacitet proširivanjem slojeva miješanja sa stohastičkim preskočnim vezama. Naša generativna arhitektura postiže izvanredne rezultate u procjeni gustoće i pokazuje se kao najprikladniji izvor sintetičkih negativnih uzoraka za gustu detekciju anomalija. Eksperimentalni rezultati pokazuju da modeli za segmentaciju nad otvorenim skupom razreda dosljedno nadmašuju postojeće metode na referentnim testovima za gustu detekciju anomalija i gusto raspoznavanje, uz zanemarivo povećanje računalnih zahtjeva.

U nastavku donosimo sažetak disertacije po poglavljima.

**Uvod**

Moderni duboki modeli imaju jake generalizacijske sposobnosti usprkos brzom zaključivanju i malom memorijskom otisku. Stoga se najnoviji napretci u robotici, kemiji i medicini čvrsto oslanjaju na duboke modele. Ipak, ove i mnoge druge aplikacije pretpostavljaju korištenje dubokih modela u kontroliranim uvjetima i ograničenom kontekstu.

U ovoj tezi analiziramo performanse dubokih modela u stvarnom svijetu koji sadrži primjere izvan trening skupa. Specifično, fokusiramo se na duboke modele za segmentaciju slika u predefinirani skup razreda. Odgovarajuće ponašanje gustih klasifikatora u ovakvom okruženju uključuje točno raspoznavanje instanci poznatih razreda te detekciju instanci nepoznatih razreda. Ovakvo ponašanje postižemo nadogradnjom gustog detektora anomalija nad standardnim klasi-

fikatorom. Specifičnost predloženog pristupa je uvođenje hibridnog detektora anomalija koji ansamblira generativni i diskriminativni pristup izgrađen povrh predtreniranog klasifikatora.

Predloženi model fino ugađamo slikama koje sadrže negativne primjere koji imitiraju testne anomalije. Negativni primjeri mogu biti uzorkovani iz pomoćnog skupa podataka ili generirani sa združeno učenim generativnim modelom. U slučaju sintetičkih negativa koristimo generativni model temeljen na normalizirajućem toku sa stohastičkim preskočnim vezama.

Provedena kvantitativna evaluacija otkriva da predloženi modeli prestižu alternativne pristupe u detekciji anomalija, segmentaciji nad otvorenim skupom razreda te procjeni izglednosti. Kvalitativna analize otkriva točnu segmentaciju poznatih dijelova scene te detekciju anomalnih dijelova scene.

**Prijašnji radovi**

Prethodni radovi relevantni za ovu tezu bave se različitim pristupima detekciji anomalija, raspoznavanju nad otvorenim skupom razreda te generativnom modeliranju. Rani pristupi detekciji anomalija na razni slike uče klasifikator na označenim podacima te detektiraju anomalije pomoću pouzdanosti klasifikacije. Bolju performansu postižu metode koje u postupak učenja uvode negativne primjere iz dodatnog raznovrsnog skupa podataka koje imitiraju testne anomalije. Klasifikatori učeni u takvom eksperimentalnom postavu generiraju predikcije s visokim stupnjem neodređenosti u negativnim primjerima. Daljnja poboljšanja se mogu postići post-hoc adaptacijom klasifikatora kao što su prorjeđivanje aktivacijskih značajki. Paralelno, generativni pristupi detekciji anomalija identificiraju anomalije temeljem procjene izglednosti.

Detekcije anomalija na razini piksela podrazumijeva da je samo dio slike anomalan. Odgovarajući pristupi stoga imitiraju testne anomalije dodavanjem negativnih podataka povrh regularnih scena. U ovakvom eksperimentalnom postavu gusti klasifikator možemo proširiti binarnim klasifikatorom koji diskriminira između regularnih i anomalnih dijelova scene. Anomalni dijelovi scene se mogu detektirati generativnim klasifikatorima ili procjenom izglednosti latentnih reprezentacija. Slično, neuspjela rekonstrukcija dijela scene može služiti kao indikator nepoznatih objekata na sceni.

Raspoznavanje nad otvorenim skupom razreda ima za cilj točnu klasifikaciju poznatih primjera te detekciju nikad prije viđenih primjera. Ovaj cilj se može postići ograničavanjem decizijske ravnine u prostoru značajki. Inicijalni pristupi modeliraju prototip svakog razreda u latentnom prostoru te odbijaju klasifikaciju ako je primjer enkodiran izvan $\varepsilon$-okoline najbližeg razreda. Alternativno, klasifikator se može komplementirati s detektora anomalija koji nadglasava odluku klasifikatora. U oba slučaja je potrebno odrediti hiperparametar koji direktno definira granicu između poznatih i nepoznatih primjera. Slično kao i u zadatku detekcije anomalija, postojanje negativnih primjera poboljšava performansu modela.

Negativni primjeri iz stvarnog skupa podataka se mogu zamijeniti sa sintetičkim negativima uzorkovanim iz združeno učenog generativnog modela. Ovakav eksperimentalni postav uklanja

pristranost prema određenim tipovima negativnih primjera viđenih tijekom učenja. Prikladan generativni model može biti brzo uzorkovan stoga se prethodni radovi često oslanjaju na generativne suparničke modele. Ipak, suparnički modeli su često nestabilni za učenje i ne pokrivaju cijelu distribuciju podataka. Alternativno, negativni primjeri se mogu konstruirati pomoću neprijateljskih perturbacija ili fraktala.

Suvremeni pristupi generativnom modeliranju se uče maksimizacijom izglednosti dostupnog skupa podataka pod distrbucijom modela. Generativne modele razlikujemo na temelju formulacija izglednosti koje mogu uključivati nenormalizirane vjerojatnosti, modeliranje latentne varijable, autoregresivnu formulaciju, zamjenu varijabli distribucije i druge.

**Gusto povezani normalizirajući tokovi**

Normalizirajući tokovi su posebna vrsta generativnih modela matematički utemeljena na formuli za zamjenu slučajnih varijabli. Formula za zamjenu slučajnih varijabli povezuje realizacije dvije slučajne varijable jednake dimenzionalnosti pomoću bijektivne diferencijabilne funkcije $f$. Pretpostavimo li da su primjeri skupa podataka realizacije slučajne varijable $\underline{\mathbf{x}}$ te da postoji latentna varijabla $\underline{\mathbf{z}}$ koja se ravna po predefiniranoj distribuciji (npr. Gaussova distribucija), tada vezu između dvije distribucije možemo modelirati nelinearnom funkcijom $\mathbf{z} = f_\theta(\mathbf{x})$. Normalizirajući tokovi stoga imaju sljedeću formulaciju: $p_\theta(\mathbf{x}) = \mathcal{N}(\underline{\mathbf{z}} = f(\mathbf{x}); \mu, \Sigma) \left| \det \frac{\partial f_\theta(\mathbf{x})}{\partial \mathbf{x}} \right|$. Učinkovita procjena izglednosti s normalizirajućim tokom zahtjeva efikasan izračun apsolutne vrijednosti determinante Jakobijana.

Generiranje novih primjera s normalizirajućim tokovima se provodi u dva koraka: uzorkovanje latentne distribucije te transformacija pomoću inverza bijekcije $f$ (to jest $\mathbf{x} = f_\theta^{-1}(\mathbf{z})$). Učinkovito uzorkovanje normalizirajućih tokova zahtjeva efikasan inverz bijektivne transformacije što uvodi dodatna ograničenja na arhitekturu modela.

Potreba za efikasnim uzorkovanjem i procjenom izglednosti je posebno naglašena kod modeliranja slika koje su u biti visokodimenzionalni podaci. Za rezultat, arhitektura normalizirajućih tokova ima značajna ograničenja te se izvodi kompozicijom niza transformacija. Tipične transformacije su konvolucija s jezgrom $1 \times 1$, bijektivna verziju normalizacije po grupi te slojevi združivanja.

Jedno ograničenje bijektivne arhitekture je fiksna dimenzionalnost latentnih reprezentacija između uzastopnih transformacija. U ovoj tezi predlažemo relaksaciju takvog ograničenja uz pomoć inkrementalnog proširivanja latentih reprezentacija u međukoracima. U praksi, to se izvodi konkatenacijom slučajnog šuma postojećim reprezentacijama, gdje su parametri distribucije šuma modelirani temeljem prethodnih latentnih reprezentacija. Navedeno unapređenje nazivamo stohastičkim preskočnim vezama.

**Hibridna segmentacija slika nad otvorenim skupom oznaka**

Segmentacija slika nad otvorenim skupom oznaka zahtjeva točnu segmentaciju instanci poznatih razreda te suzdržavanje od odluke u nepoznatim dijelovima slike. Nepoznati dijelovi slike

su u biti semantičke anomalije, stoga ovaj zadatak možemo riješiti kombiniranjem gustog klasifikatora i detektora anomalija.

Prethodni radovi koriste detektore anomalija koje se mogu grubo kategorizirati u diskriminativne i generativne pristupe. Diskirminativni i generativni detektori anomalija optimiraju različite gubitke te stoga čine različite greške čak i kad dijele latentne reprezentacije. Ova teza predlaže ansambliranje diskriminativnog i generativnog pristupa izgrađenih povrh standardnog klasifikatora u jedinstveni hibridni detektor anomalija. Generativni detektor anomalija izgrađujemo povrh standardnog klasifikatora reinterpretacijom eksponenciranih logita kao nenormalizirane zajedničke distribucije ulaza i razreda. Marginalizacijom preko svih razreda osiguravamo nenormaliziranu izglednost ulazne slike. Diskriminativni detektor anomalija gradimo nad standardnim klasifikatorom kao dodatni binarni klasifikator povrh predlogita. Rezultirajući model ima tri izlaza: klasifikator poznatih razreda, nenormalizirana izglednost te aposteriorna vjerojatnost skupa podataka.

Adekvatno ponašanje modela za raspoznavanje postižemo finim ugađanjem na negativnih primjerima. Negativni primjeri se uzorkuju iz pomoćnog skupa podataka te lijepe povrh regularnih scena kako bi se dobile scene miješanog sadržaja. Ova teza predlaže zamjenu negativnih primjera iz pomoćnog skupa s umjetnim negativima uzorkovanih s generativnim modelom. Specifično, umjetne negative uzorkujemo iz združeno učenog normalizirajućeg toka sa stohastičkim preskočnim vezama.

**Metodologija**

Evaluiranje procjene izglednosti, segmentacije slika nad otvorenim skupom oznaka te guste detekcije anomalija zahtjeva adekvatne skupove podataka koje dijelimo u tri kategorije. Skupovi podataka s malim slikama rezolucije do $64 \times 64$ piksela se koristimo za procjenu izglednosti s generativnim modelima. Primjer ovakvog skupa podataka je CIFAR10 ili ImageNet32. Skupovi podataka sa općenitim scenama uključuju slike raznovrsnih objekata. Primjer ovakvog skupa je MS COCO, koji sadrži guste oznake za preko sto različitih razreda. Ovaj skup koristimo pri validaciji performanse modela za raspoznavanje nad otvorenim skupom oznaka. Konačno, skupovi sa prometnim scenama sadrže slike vožnje iz perspektive vozača automobila. Primjer ovakvnog skupa podataka je Fishyscapes koji sadrži semantički bogate slike visoke rezolucije s izvandistribucijskim primjerima. Slike vožnje koristimo za gustu detekciju anomalija te raspoznavanje nad otovrenim skupom razreda.

Kvantitativnu procjenu performanse na navedenim skupovima podatak provodimo standardnim metrikama kao što su prosječna preciznost, površina ispod ROC krivulje i druge. Dodatno, predlažemo novu metriku pod nazivom open-IoU koja uključuje lažne pozitive i lažne negative u anomalnim pikselima u procjenu segmentacijske točnosti modela. Konačno,poglavlje sadrži glavne implementacijske detalje potrebni za reproduciranje provedenih eksperimenata.

**Rezultati**

Kvantitativni rezultati validiraju arhitekturu normalizirajućeg toka DenseFlow na standardnim testnim skupovima CIFAR10, CelebA i ImageNet. DenseFlow postiže bolju procjenu izglednosti od alternativnih arhitektura normalizirajućeg toka.

Predloženi hibridni detektor anomalija DenseHybrid je validiran na skupovima Fishyscapes i SegmentMeIfYouCan te postiže bolju performansu od prethodnih pristupa. Slično, segmentacijski model s DenseHybrid detektorom anomalija postiže najbolje rezultate u segmentaciji nad otvorenim skupom oznaka na skupovima podataka StreetHazards i COCO. Daljnja analiza pokazuje da DenseHybrid zahtjeva minimalne računske zahtjeve.

Kvalitativni rezultati pokazuju slike generirane s predloženim normalizirajućim tokom, primjere sintetičkih negativa te primjere segmentacije za slike iz različitih skupova podataka.

**Zaključak i budući rad**

Ova disertacija predlaže novi pristup segmentaciji slika nad otvorenim skupom oznaka koji kombinira standardne guste klasifikatore i hibridni detektor anomalija. Predloženi hibrindni pristup agregira generativni i diskirminativni detektor anomalija u jedinstveni detektor. Gusti klasifikator s hibridnim detektorom anomalija je potrebno fino ugoditi na slikama koje sadrže izvandistribucijske primjere preuzete iz dodatnog skupa podataka ili generirane uz pomoć združeno učenog generativnog modela. U potonjem slučaju, umjetni negativni podaci su uzorkovani pomoću normalizirajućeg toka sa stohastičkim preskočnim vezama. Evaluacija predloženih modela pokazuje poboljšanje performanse u usporedbi s alternativne pristupa na različitim skupovima podataka.

Budući rad uključuje daljna poboljšanja segmentacije slika nad otvorenim skupom oznaka, testiranje predložene metode na recentnim pristupima segmentaciji koji koriste raspoznavanje na razini maski te adaptacija recentnih generativnih modela kao izvor negativnih podataka.

**Dodatak**

Dodatak sadrži detaljno raspisane dokaze za donju granicu izglednosti kod gusto povezanih normalizirajućih tokova, detalje dvodimenzionalnog skupa podataka korištenog u ilustraciji četvrtog poglavlja, detaljne izvode gubitaka korištenih za fino ugađanje hibridnog detektora anomalija, te proširene razulate DenseFlow arhitekture.

**Ključne riječi**: Semantička segmentacija, Segmentacija nad otvorenim skupom razreda, Raspoznavanje nad otvorenim skupom razreda, detekcija anomalija, detekcija izvandistribucijskih primjera, umjetni negativni podaci, generativni modeli, normalizirajući tok

# Contents

# Chapter 1

# Introduction

Extensive generalization capabilities, fast inference and small memory footprint of contemporary neural networks steadily expand the horizon of real-world applications. Recent advances in robotics [1], chemistry [2] and medicine [3] heavily rely on deep learning technology. However, many of these applications assume deployment in closed environments with limited variability. Conversely, open-world applications such as transportation [4, 5], present a challenge for contemporary deep models due to the diversity of the environment. Moreover, existing evaluation protocols related to the open world deployments [4, 6, 7] focus on unrealistic setups that overlook possible hazards of the real world. For instance, most semantic segmentation datasets [4, 7] annotate only instances of known classes, while ignoring and deeming the ramaining content as out of scope.

This thesis aims to go beyond closed-set benchmarks and evaluate deep recognition models in the presence of irregular test examples that deviate from the training distribution. In particular, we focus on dense prediction context and a specific kind of out-of-distribution examples known as semantic anomalies. Semantic anomalies are instances of classes that do not belong to the training taxonomy [8]. To effectively handle anomalous test data, we enable our segmentation models to detect instances of previously unseen classes, *i.e.* semantic anomalies, while correctly classifying the instances of the inlier classes. The described task is commonly referred to as open-set segmentation.

We extend the standard segmentation by incorporating the ability to withhold the semantic decision by complementing the standard closed-set classification with a dense semantic anomaly detector [9, 10, 11]. This approach aligns with several prior works [12, 13, 14, 15] which propose anomaly detectors generally categorized into generative and discriminative approaches. We hypothesize that the two categories of anomaly detectors exhibit different failure modes, even when built atop the same feature representations [16]. If our hypothesis holds, combining the two approaches into a hybrid anomaly detector could yield a more accurate detector than either component alone.

Given the necessity for computationally efficient inference in practical applications, we construct a lightweight dense hybrid anomaly detector that introduces minimal computational overhead over the standard dense classifier. Specifically, the generative component of our hybrid anomaly detector is developed through a reinterpretation of classifier logits [17], while the discriminative component is formulated as an auxiliary outlier detection head [10]. We denote the resulting method for anomaly detection as *DenseHybrid*. Our method facilitates open-set inference by implementing straightforward and compact upgrades to pre-trained dense classifiers [18], typically with negligible impact on inference time. Figure 1.1 illustrates an example of open-set inference with DenseHybrid. Given an input image, we produce closed-set semantic predictions alongside a dense anomaly map. The anomaly map aggregates generative and discriminative anomaly scores. The final open-set output is recovered by overriding closed-set predictions in pixels identified as anomalous.

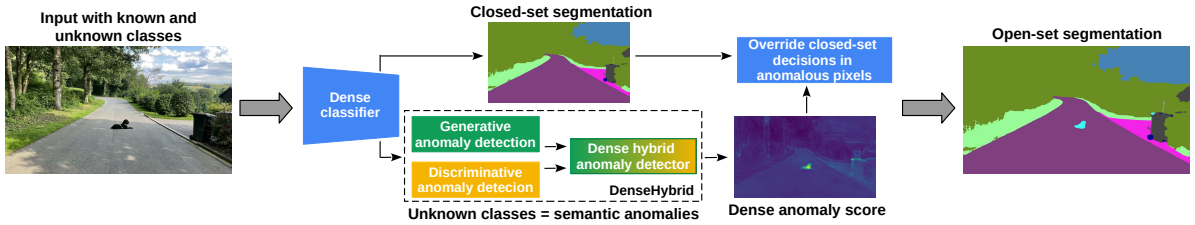


**Figure 1.1:** Open-set segmentation simultaneously classifies known scene parts and identifies unknown classes (highlighted in cyan). Our approach exploits the fact that unknown classes are semantic anomalies [8]. Hence, we construct a dense hybrid anomaly detector and use it to detect anomalous pixels. Our hybrid anomaly score identifies pixels as unknown visual concepts by efficient ensembling of generative and discriminative predictions.

The DenseHybrid approach can upgrade any pre-trained dense classifier with open-set recognition capability. We propose a fine-tuning procedure that uses real or synthetic negative data to simulate test anomalies. The negative data is overlaid onto inliers images to create mixed-content training images [12]. These images are then used to train the open-set model by optimizing appropriate objectives. In the case of synthetic negative data, we jointly train a normalizing flow capable of generating dataset-specific samples with varying spatial dimensions. Specifically, we use *DenseFlow*, a normalizing flow with stochastic skip connections [19].

The resulting models are evaluated on various test scenarios, including general images and application-specific road-driving scenes. We quantify dense open-set recognition performance in the presence of outliers with the novel open-IoU metric that penalizes both semantic false positives at outliers and semantic false negatives at inliers. DenseHybrid consistently outperforms alternative approaches both in semantic anomaly detection and open-set segmentation. Altogether, this thesis proposes the following contributions:

1. A hybrid anomaly score that ensembles the discriminative and the generative component
2. An algorithm to learn a translationally equivariant model of the neighborhood density, which avoids the expensive estimation of the normalizing constant by minimizing the

likelihood of negative training data

3. Elements of a differentiable module that creates artificial negative examples that are used for training an open-set semantic segmentation model
4. Generative normalizing flow architecture with stochastic skip connections

These contributions are elaborated in the rest of the thesis with the following structure. The second chapter, titled "Related work", begins by revisiting previous studies that deal with image-wide and pixel-wise anomaly detection. The chapter continues by reviewing open-set recognition and the use of synthetic data in open-set setups. The chapter concludes with an in-depth overview of existing generative models.

The third chapter, titled "Densely connected normalizing flows", starts with a brief review of the change of variables formula, the fundamental mathematical concept behind normalizing flows. The chapter proceeds by describing the standard building blocks of bijective flows and their adaptations for image data. The final section introduces stochastic skip connections for normalizing flows, which is the first contribution of this thesis.

The fourth chapter, titled "Hybrid open-set segmentation of images", introduces open-set segmentation by fusing a closed-set segmentation model with a dense hybrid anomaly detector. The hybrid anomaly detector fuses discriminative class posterior with dense unnormalized likelihood, which is the second contribution of this thesis. The proposed unnormalized likelihood takes an equivariant form particularly designed for dense prediction, which is the third contribution of this thesis. The chapter proceeds by elaborating fine-tuning of the open-set model with real negative data. The real negative data can be replaced with synthetic negative data, which is the fourth contribution of this thesis.

The fifth chapter, "Methodology" explains the experimental setup used to validate the proposed contributions. The chapter describes datasets and benchmarks used in open-set and density estimation experiments, followed by performance metrics for model evaluation. The chapter concludes with the main implementation details relevant for reproducibility of our results.

The sixth chapter, titled "Results", includes experimental results for density estimation, per-pixel anomaly detection and open-set segmentation. The chapter presents quantitative performance comparisons with existing baselines and previous works. Ablation studies validate the proposed methodological contributions while computational analysis covers the practical aspects of the proposed contributions. The chapter concludes with qualitative results.

The seventh chapter, titled "Conclusion and outlook", concludes this thesis while the eighth chapter "Appendix" lists detailed proofs and extended derivations.

# Chapter 2

# Related Work

This chapter revisits the related work in anomaly detection, open-set segmentation and generative modeling. Section 2.1 reviews image-wide anomaly detection. Section 2.2 considers anomaly detection at the pixel level. Section 2.3 outlines prior research in open-set recognition. Section 2.4 describes the utilization of generative models for synthetic training data. Section 2.5 describes further advances from open-set recognition towards open worlds. Finally, Section 2.6 revisits contemporary generative modeling approaches.

Detecting observations that deviate from some notion of regularity is a decades-old problem [20]. In the age of big data, regularity is often specified by the available examples, commonly referred to as inliers, while the irregular data is then referred to as outliers. Outliers significantly differ from the inliers in some way and are often also referred to as anomalies, out-of-distribution data, or novelties [8]. Throughout this work, we will use these terms interchangeably. Modern research in anomaly detection considers different setups with a broad set of anomalies [21, 22]. We will focus on methods that aim at detecting semantic anomalies, *i.e.* instances of classes outside the training taxonomy. In this setup, the set of inlier classes is known beforehand and the available dataset is annotated. The full spectrum of research in anomaly detection can be found in surveys [8, 23]

## 2.1 Image-wide anomaly detection

Given a labeled inlier dataset, early image-wide approach [24] trains K-way classifier and utilizes max-softmax probability to detect anomalous inputs. The succeeding approach ODIN [25] introduced anti-adversarial input perturbations in order to improve the detection performance. Anomaly detection with predictive uncertainty can be further improved through Bayesian formulation [26] and ensembling of multiple models [27]. Instead of prediction confidence, max-logit score (MLS) [15] detects anomalous inputs based on logit scores, while GradNorm [28] considers the norm of the gradient with respect to model parameters [29]. These approaches

had limited success since the model training lacked the notion of anomaly.

More encouraging performance has been attained by learning on surrogate anomalies that we denote as negative training data. [10, 30, 31, 32]. The negative examples are commonly sourced from a broad auxiliary dataset in order to account for the sheer diversity of all possible test-time anomalies [31]. Initial approaches require high entropy predictions in negative training data [30, 31]. Succeeding work [32] fits an energy function to the logits of the discriminative model and requires low energy in inliers and high energy in negative samples. During the inference, anomalous samples are detected according to the energy score.

Further performance improvements have been attained by carefully sampling negative training examples [33, 34]. The seminal work ATOM [33] mines informative negative samples by ranking them according to the OOD score, *i.e.* the confidence of K+1-th class. Similarly, POEM [34] retrieves informative negative samples according to the distance between the inliers and candidate negatives in the feature space, which is assumed to be informative. This idea is further advanced in DOS [35] by introducing a sampling strategy that also accounts for the diversity in negative samples. In the context of negative samples, the diversity corresponds to pairwise distances between the sampled negatives. Perturbing samples in the direction of gradient w.r.t negative loss can further increases the variety of training negatives [36].

The trained model can be further post-processed in order to improve outlier detection performance. For instance, ReAct [37] observes that outliers give rise to feature representations with large norm in pre-logit space Thus, ReAct truncates the pre-logit features to predefined treshold and detects anomalies based on different anomaly scores, *i.e.* softmax confidence [24] or free-energy [32]. This operation limits the effects of large feature representations for some inputs. ASH [38] further advances this idea by completely removing some activations, while DICE [39] sparsifies the model weights to achieve the same effect. Sparse activations prevent aggregation of uninformative features in the computation of the logits. Alternatively, [40] introduces a memory module that collects inlier feature representations. The collected memory is then used for detecting anomalous inputs based on the distance from nearest inlier neighbors.

Another line of work detects anomalies by estimating the data likelihood. In this setup, a low likelihood of a given sample would indicate anomalous input. Surprisingly, this research reveals that anomalous images may give rise to a higher likelihood than inliers [41, 42, 43]. This problem can be alleviated by modeling condensed representations of the input, *i.e.* features in low dimensional space [13]. Thus, interesting approaches [13, 44] fit a probabilistic density estimator to features of a discriminative model.

## 2.2 Pixel-wise anomaly detection

Natural images include objects within a certain context. The goal of pixel-wise anomaly detection is to segment anomalous image parts, which can include anomalous objects, context, or both. Image-wide anomaly detectors can be adapted for dense prediction with variable success. Some image-wide approaches are not applicable in dense prediction setups [45], while others do not perform well [24] or involve excessive computational complexity [25, 27]. On the other hand, discriminative training with negative data [12, 30, 31] is easily ported to dense prediction.

Early dense discriminative anomaly detector [10, 12] appends an additional OOD head that differentiates inliers from outliers and trains the additional head on mixed-content images. The mixed-content images are obtained by pasting negative content (e.g. ImageNet1k, COCO, ADE20k) over regular training images. Alternatively, outlier pixels can be detected based on max-logit value [15] that can be standardized through post-processing for further performance improvements [46].

Similarly, anomalous scene parts can be detected by considering prediction uncertainty. A seminal approach [26] models aleatoric and epistemic uncertainty through Bayesian deep learning. Alternatively, [47] captures uncertainty by explicitly parameterizing a prior over predictive distributions. In both cases, high uncertainty predictions indicates to anomalous scene parts. Finally, [11] detects outliers based on prediction entropy of a model trained on real and negative training data.

Generative dense anomaly detectors detect outlier pixels by estimating the density. EmbeddingDensity [9, 13] models the likelihood of feature activations at different stages of the trained model. However, this approach may be sensitive to feature collapse [48]. Interesting approach GMMSeg [14] constructs a generative classifier and detects anomalies based on per-class likelihood. Finally, PEBAL [49] models the energy surface through abstention learning.

Another line of work utilizes image resynthesis for dense anomaly detection. In this context, pixel-wise anomaly detectors can be implemented according to the learned dissimilarity between the input and resynthesized images [50, 51, 52]. Regions with significant deviation from the resynthesised image indicate anomalous content. The resynthesis can be performed by a generative model conditioned on the predicted labels. However, this approach is suitable only for uniform backgrounds such as roads [50] since reconstruction is an ill-posed problem. Furthermore, conditional generation involves a large computational overhead that precludes many real-world applications dependent on real-time inference. The burden of resynthesis can be lessened through reconstruction from low-dimensional latent space [53]. Furthermore, image resynthesis can be circumvented by comparing the texture of potentially anomalous objects with the texture of the surrounding road [54].

Different from all previous work, we propose a hybrid anomaly detector for dense prediction

models that fuses discriminative and generative anomaly scores built atop a dense classifier. In comparison with previous approaches that build on inlier posterior [10, 30, 31], our method introduces synergy with unnormalized likelihood evaluation. In comparison with approaches that recover dense likelihood [9], our method introduces joint hybrid end-to-end training and efficient joint inference together with standard semantic segmentation. Our method is related to joint energy-based models [17, 55], since we also reinterpret logits as unnormalized joint likelihood. However, previous energy-based approaches have to backpropagate through the intractable normalization constant and are therefore unsuitable for large resolutions and dense prediction. Our method avoids model sampling by contrastive training on inlier and negative data.

## 2.3 Open-set recognition

Open-set recognition requires the identification of inlier classes while withholding (or rejecting) the decision for instances of unknown classes [56]. Figure 2.1 compares the task of open-set recognition with the two closest tasks, multiclass classification and semantic anomaly detection. Different than the multiclass classification, open-set recognition necessitates the detection of classes unseen during the training. Different from anomaly detection, open-set recognition requires further identification of inlier content according to the set of known classes.

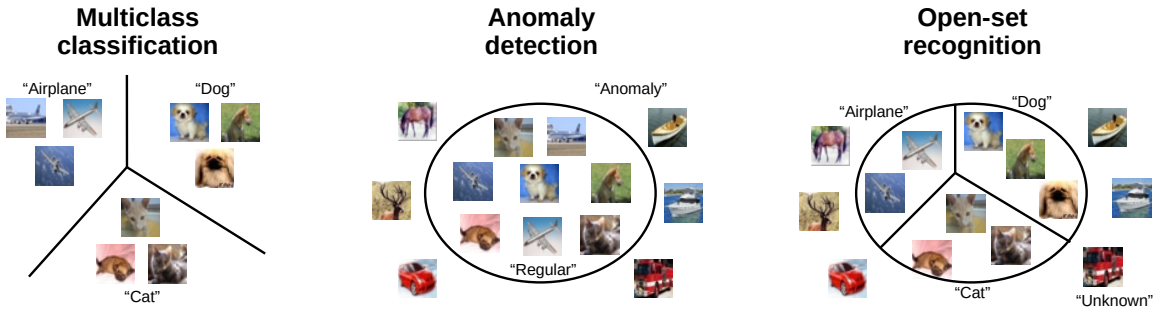


**Figure 2.1:** The task of open-set recognition requires correct classification of inlier content and detection of instances unseen during the training [56].

Withholding the decision in instances of unknown can be done by restricting the shape of the decision boundary [57, 58]. A seminal approach [57] learns class centers in the embedding space and rejects the decision in instances embedded far from the nearest class prototype. DML [59] further scales this approach for dense recognition. Still, these approaches cannot deal with outliers that are embedded near the class centers, which may be the case with deep feature extractors. Feature collapse can be alleviated by learning reciprocal points [60]. In this training setup, class prototypes are pushed far from representations of other semantic content, *i.e.* instances of other classes and the negative examples. The quality of embedding space can

be further improved by utilizing adversarial training examples [61]. OpenHybrid [45] fits the normalizing flow on feature representations of inlier content in order to withhold the decision for instances that yield unlikely feature representations. As with many machine learning tasks, open-set performance can be drastically improved by supplying more capacity through deeper architectures [62].

The rejection mechanism can alternatively be formulated by complementing the classifier with a thresholded semantic anomaly detector [24, 25, 63]. In this case, the binary decision of the semantic anomaly detector overrides closed-set predictions of the classifier. The resulting output is again K+1-way recognition map. More details about open-set approaches can be found in the recent review [64].

Most open-set approaches quantify performance by separate evaluation of closed-set recognition and anomaly detection [9, 24, 65, 66]. However, such practice does not reveal degradation of discriminative predictions due to errors in anomaly detection [67, 68]. This is especially pertinent to dense prediction where we can observe inlier and outlier pixels in the same image. Recent work proposes a solution for the related problem of semantic segmentation in adverse conditions [69]. Their uncertainty-aware UIoU metric takes into account prediction confidence as measured by the probability of the winning class. However, UIoU assumes that each pixel belongs to one of the known classes, which makes it inapplicable for open-set setups. Different than all previous work, our open-IoU metric specializes for open-set segmentation in the presence of outliers. It takes into account both false positive semantic predictions at outliers as well as false negative semantic predictions due to false positive anomaly detection. Furthermore, the difference between mIoU and open-mIoU reveals the performance gap of the open-set setup.

## 2.4 Synthetic data in open-set recognition

Existing datasets for open-set recognition often provide instances of some classes that transcend the training taxonomy and can be used for the training. These training data points are usually referred to as negative training data [31] or known unknowns [56]. However, training on real-world negatives will typically introduce a bias towards the detection of a particular subset of all possible anomalies. Consequently, the performance metrics will be likely over-optimistic. depending on the prevalence of such content in the test data.

Recent seminal approaches [70, 71] utilize synthetic negative data produced by a jointly trained generative adversarial network instead of training negatives. The corresponding GAN is trained to generate inlier data that give rise to low recognition scores for each known class [71]. However, GANs offer only limited distribution coverage [48]. Consequently, they are unlikely to span the whole space of possible unknowns. Thus, succeeding works mix real and synthetic negative examples [72].

Distributional coverage can be improved by replacing GANs with generative models that optimize likelihood [48]. Our context calls for efficient approaches with fast sampling since joint training requires sample generation on the fly. This puts the autoregressive and energy-based models at a disadvantage, except in the context of very small images. Normalizing flows are a great candidate for this role due to fast training and fast sample generation at different resolutions [73, 74]. Instead of targeting negative data, a generative model can also target negative features [72]. This can be done by modelling inlier features and sampling synthetic anomalies from low-likelihood regions of feature space [44, 75].

Synthetic negative data can also be crafted by leveraging adversarial perturbations [76]. In this setup, adversarially perturbed input examples are used as a proxy for negative data. Similarly, [55] uses negative samples from an implicit energy-based generator in the standard multiclass classifier. Alternatively, negative examples can be constructed by mixing inlier images with a precomputed set of fractals [77]. Interestingly, applying multiple image augmentations to inlier images results in a sufficient proxy for real negative data [78]. In the dense prediction context, one can also crop convex polygons of inlier content and paste them on different spatial locations within the same image [53].

## 2.5 Beyond open-set recognition

Once the instances of unknown classes are detected, we can further cluster them to form new semantic classes. This can be done by incrementally increasing the set of known classes [79, 80]. For instance, features from uncertain image regions could be clustered into new potential classes by pseudolabeling followed by fine-tuning [80]. Alternatively, novel classes can be discovered end-to-end starting from self-supervised feature representations [81, 82].

Novel semantic classes can be defined by a small set of examples that are readily available in the data. This setup corresponds to few-shot learning [83]. Similarly, one can incorporate meta-information about novel classes, which gives rise to zero-shot learning [84]. We direct the reader to [85] for an exhaustive analysis of pros and cons of low-shot learning. Note that all of these approaches are still unable to compete with supervised learning on standard datasets.

## 2.6 Generative modeling

The main goal of generative modeling is to approximate data distribution $p_D$, loosely defined by a set of i.i.d samples $\mathscr{D}$, with a model distribution $p_\theta$. The set of samples $\mathscr{D}$ is of finite size $N$ and $\theta \in \mathbb{R}^d$ represents learnable parameters. The optimal set of parameters $\theta^*$ is commonly

obtained by optimizing the following objective:

$$\theta^* = \underset{\theta\in\Theta}{\operatorname{argmin}}\, \mathrm{KL}(p_D||p_\theta) = \underset{\theta\in\Theta}{\operatorname{argmin}}\, \mathbb{E}_{\mathbf{x}\sim p_D}[-\ln p_\theta(\mathbf{x})] \approx \underset{\theta\in\Theta}{\operatorname{argmin}}\, \frac{-1}{N}\sum_{i=1}^{N} \ln p_\theta(\mathbf{x}^i)\,. \tag{2.1}$$

Here, KL stands for Kullback–Leibler divergence between the two distributions. The first equality in (2.1) holds since the difference between the two objectives is constant and equal to the negative entropy of $p_D$. The third objective approximates the expectation with the mean over an i.i.d dataset $\mathscr{D}$. We distinguish different families of generative models depending on their definition of $p_\theta$. We next briefly describe many interesting formulations of the model distribution.

**Energy-based models** [86] define model distribution via the Boltzmann distribution:

$$p_\theta(\mathbf{x}) := \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)}, \qquad Z(\theta) = \int \exp(-E_\theta(\mathbf{x}))\, d\mathbf{x}\,. \tag{2.2}$$

Note that $E_\theta : \mathscr{X} \to \mathbb{R}$ denotes a scalar energy function defined using a deep neural network. Training of energy-based models necessitates approximation of the intractable normalization constant $Z$, as detailed in [87]. Generating samples using EMBs necessitates iterative MCMC sampling that can be carried out according to the Langevin dynamics [88]. In practice, such sample generation requires computing the gradient of the energy function w.r.t input ($\nabla_{\mathbf{x}} E_\theta$) at every iteration step [87]. Slow convergence of MCMC algorithms in high dimensional spaces, makes sample generation with energy-based models notoriously slow.

**Autoregressive models** [89] assume autoregressive factorization of the model distribution:

$$p_\theta(\mathbf{x}) := \prod_{i=1}^{\dim(\mathbf{x})} p_\theta(\mathbf{x}_i|\mathbf{x}_{<i})\,. \tag{2.3}$$

Here, $\mathbf{x}_i$ denotes the $i$-th element of the vector (or vectorized tensor) $\mathbf{x}$, while all elements before $i$-th element are denoted with $< i$. Conditional distribution $p_\theta(\mathbf{x}_i|\mathbf{x}_{<i})$ is usually implemented with a deep model to obtain sufficient modeling capacity. Autoregressive models are easily trained by likelihood maximization but require sequential sampling with $\dim(\mathbf{x})$ steps.

**Variational autoencoders** [90] combine variational inference and autoencoders to model the relationship between the latent random variable $\underline{\mathbf{z}}$ and the observed random variable $\underline{\mathbf{x}}$. The log distribution $\ln p_\theta(\mathbf{x})$ can be modeled as:

$$\ln p_\theta(\mathbf{x}) = \ln \int q_\psi(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\psi(\mathbf{z}|\mathbf{x})}\, d\mathbf{z} \geq \mathbb{E}_{\mathbf{z}\sim q_\psi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q_\psi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]\,. \tag{2.4}$$

The first equality introduces marginalization of $\mathbf{z}$, while the second inequality follows from the direct application of Jensen inequality. Here, encoder models $q_\psi(\mathbf{z}|\mathbf{x})$ while decoder models $p_\theta(\mathbf{x}|\mathbf{z})$. Maximizing the log-likelihood of dataset samples effectively minimizes reconstruc-

tion error (first term) and aligns $q_\psi(\mathbf{z}|\mathbf{x})$ with latent prior (second term). Both encoder and decoder can be jointly learned with gradient descent by utilizing the reparameterization trick [90]. For example, if variational posterior $q$ is a normal distribution, the encoder predicts the mean and covariance of the posterior $q$. Sampling the posterior $q$ then proceeds by sampling the standard normal and transforming the sample according to the predicted distribution parameters. Producing new samples with variational autoencoder can be done by sampling the latent prior and decoding the sampled $\mathbf{z}$ into the corresponding $\mathbf{x}$ using the decoder.

**Diffusion-based models** [91] define the model distribution using a $T$ step Markov chain with latents $\mathbf{x}_1, \ldots, \mathbf{x}_T$ of the same dimensionality as the input:

$$p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T} = \int p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\, d\mathbf{x}_{1:T}\,. \tag{2.5}$$

Here, $p_\theta(\mathbf{x}_{0:T})$ represents joint distribution commonly referred to as the *reverse process*, $p(\mathbf{x}_T)$ is an isotropic Gaussian distribution, and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is a Gaussian with mean and covariance computed using a deep model parameterized with $\theta$. Different from previous models, diffusion-based models are trained to reverse the *forward process* $q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$. Every step $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I})$ with predefined variance schedule $\beta_t$ essentially adds Gaussian noise to the input example. Introducing the described forward process as variational distribution in (2.5) yields the following evidence lower bound:

$$\ln p_\theta(\mathbf{x}_0) \geq \mathbb{E}_q \left[ \ln \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] = \mathbb{E}_q \left[ \ln p(\mathbf{x}_T) + \sum_{t=1}^{T} \ln \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right]. \tag{2.6}$$

**Generative adversarial networks** [92] implicitly learn the data distribution through a minimax game of two players - discriminator $D$ and generator $G$:

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\ln D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\ln(1 - D(G(\mathbf{z})))] \tag{2.7}$$

The discriminator aims to differentiate real dataset examples from artificial examples produced by the generator. The generator aims to produce samples that trick the discriminator. Given that the discriminator is optimal, the generator objective boils down to the minimization of Jensen-Shannon divergence between the data distribution and the generator distribution [92]. In practice, both the discriminator and generator are deep models. Consequently, optimizing the minimax objective (2.7) over a large parameter space may be unstable. As a result, the network generator may produce similar samples without the full coverage of the data distribution [48]. Poor coverage of the inlier distribution is commonly referred to as mode collapse. A more comprehensive overview of deep generative models can be found in [93].

# Chapter 3

# Densely connected normalizing flows

This chapter considers a family of generative models that are known as normalizing flows. Normalizing flows are mathematically grounded on change of variables, as described in Section 3.1. Section 3.2 presents the standard building blocks of bijective flows. This thesis applies bijective flows for generative modelling of natural images, as detailed in Section 3.3. Finally, Section 3.4 proposes a novel family of bijective flows that can gradually increase the latent dimensionality through stochastic skip connections.

**Notation.** This thesis deals with natural images of width $W$ and height $H$. The input images $\mathbf{x} \in \mathscr{X}$ are modeled with a random tensor $\underline{\mathbf{x}}$ of dimensions $C \times H \times W$. Thus, a pixel at the location $(i, j)$ is modeled by the corresponding $C$-dimensional random vector $\underline{\mathbf{x}}^{ij}$. The realization of a random variable is denoted by omitting the underline while the spatial locations are often omitted for brevity. Thus, $p(\mathbf{x})$ is a shortcut for $p(\underline{\mathbf{x}} = \mathbf{x})$.

## 3.1 Generative modeling via change of variables

The main goal of generative modeling is to learn a model $p_\theta$ that approximates the unknown data distribution $p_D$ loosely defined by a finite set of realizations $\mathscr{D}$. A well-trained model could then produce new samples by sampling the learned model distribution $p_\theta$.

Given a set of training samples $\mathscr{D} = \{\mathbf{x}^i\}_{i=1}^N$ of size $N$, we train generative models by minimizing the KL divergence between $p_D$ and $p_\theta$. The optimal set of parameters $\theta^*$ is obtained by solving the following optimization objective:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \, \mathrm{KL}(p_D || p_\theta) = \underset{\theta \in \Theta}{\operatorname{argmin}} \, \mathbb{E}_{\mathbf{x} \sim p_D(\underline{\mathbf{x}})}[-\ln p_\theta(\mathbf{x})] \approx \underset{\theta \in \Theta}{\operatorname{argmin}} \, \frac{-1}{N} \sum_{i=1}^{N} \ln p_\theta(\mathbf{x}^{(i)}) \,. \quad (3.1)$$

### 3.1.1 Change of variables

In the case of normalizing flows [94, 95] we define the model distribution $p_\theta$ by employing the change of variables formula. Let $\underline{\mathbf{x}}$ and $\underline{\mathbf{z}}$ be two continuous random vectors with probability density functions $p(\underline{\mathbf{x}})$ and $p(\underline{\mathbf{z}})$. Both random vectors have realizations in $\mathbb{R}^d$. Given a differentiable bijective function $f : \mathbb{R}^d \to \mathbb{R}^d$, we can rephrase $p(\underline{\mathbf{x}} = \mathbf{x})$ using $p(\underline{\mathbf{z}})$ as:

$$p(\underline{\mathbf{x}} = \mathbf{x}) = p(\underline{\mathbf{z}} = f(\mathbf{x})) \left| \det \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right| . \tag{3.2}$$

Here $\det(\cdot)$ represents matrix determinant, $|\cdot|$ returns absolute value while $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ denotes the Jacobian of $f$ evaluated at $\mathbf{x}$. Consequently, the density of $p(\underline{\mathbf{x}})$ at $\mathbf{x}$ corresponds to the density of $p(\underline{\mathbf{z}})$ at $f(\mathbf{x})$ adjusted by the change of volume factor $\left| \det \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right|$. A more comprehensive description of the change of variables formula can be found in textbooks such as [96].

Figure 3.1 shows a two-dimensional example where $p(\underline{\mathbf{x}})$ is a Gaussian mixture of three components and $p(\underline{\mathbf{z}})$ is a multivariate Gaussian. The mapping function $f : \mathscr{X} \to \mathscr{Z}$ is nonlinear since *i*) linear perturbation of normally distributed data is also normally distributed, and *ii*) normal distributions can not describe multimodal distributions such as the one on the left figure. We will next consider convenient parameterizations of the function $f$ and distribution $p(\underline{\mathbf{z}})$ in order to build a generative model of the original data.

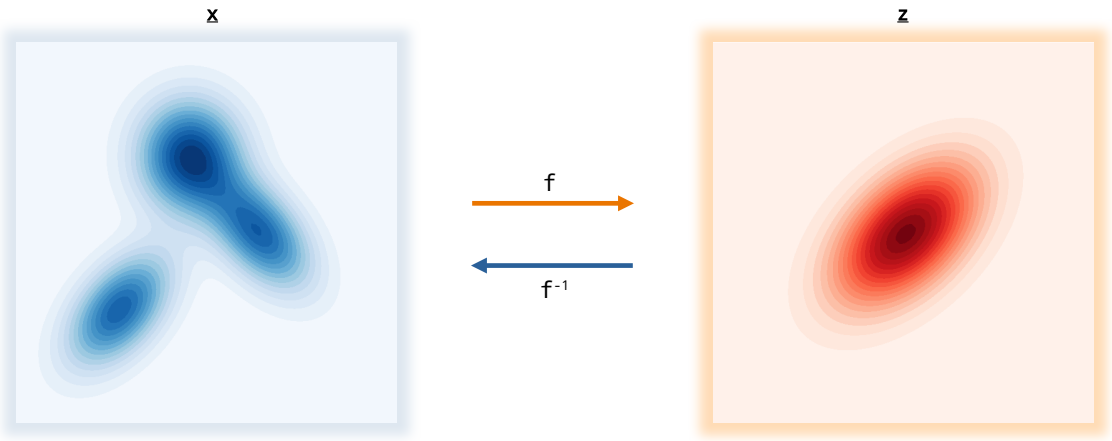


**Figure 3.1:** The change of variables formula involves two probability distributions $p(\underline{\mathbf{x}})$ and $p(\underline{\mathbf{z}})$ via a differentiable bijective function $f$.

### 3.1.2 Normalizing flows

We build normalizing flows by parameterizing the function $f$ with a set of learnable parameters $\theta$ and by assuming $p(\underline{\mathbf{z}})$ is a multivariate Gaussian distribution whose parameters can be either

learnt or preselected. The normalizing flow $p_\theta$ then corresponds to:

$$p_\theta(\mathbf{x}) = \mathcal{N}(\underline{\mathbf{z}} = f(\mathbf{x}); \mu, \Sigma) \left| \det \frac{\partial f_\theta(\mathbf{x})}{\partial \mathbf{x}} \right|. \tag{3.3}$$

Modeling natural data with normalizing flows requires mapping a highly complex data distribution $p(\underline{\mathbf{x}})$ to a simple Gaussian $p(\underline{\mathbf{z}})$. This can be done successfully only if the mapping function $f$ has sufficient capacity. Hence, we next focus on the design of function $f$.

A composition of bijective functions is a bijective function. Thus, we consider to decompose $f_\theta$ into a sequence of $K$ bijections $f_i$ with its subsets of parameters $\theta_i$. This composition can be visualized as:

$$\mathbf{z}_0 \overset{f_1}{\longleftrightarrow} \mathbf{z}_1 \overset{f_2}{\longleftrightarrow} \mathbf{z}_2 \overset{f_3}{\longleftrightarrow} \cdots \overset{f_{i-1}}{\longleftrightarrow} \mathbf{z}_i \overset{f_i}{\longleftrightarrow} \cdots \overset{f_K}{\longleftrightarrow} \mathbf{z}_K, \quad \vec{z}_K \sim \mathcal{N}(0, \mathrm{I}). \tag{3.4}$$

Here, $\mathbf{z}_0$ is essentially the sample $\mathbf{x}$ drawn from the data distribution $p_D$. The deep model $f_\theta$ maps samples $\mathbf{z}_0$ to their normally distributed counterparts $\mathbf{z}_K$, This will allow for evaluation of the probabilistic density $p(\mathbf{z}_0)$ as well as sampling the learned distribution subject to some additional requirements on $f_\theta$.

Following the change of variables formula, log-likelihoods of consecutive random variables $\mathbf{z}_i$ and $\mathbf{z}_{i+1}$ can be related through the Jacobian $J_{i+1}$ of the corresponding transformation $f_{i+1}$:

$$\ln p(\mathbf{z}_i) = \ln p(\mathbf{z}_{i+1}) + \ln |\det J_{i+1}|. \tag{3.5}$$

The relation (3.5) can be seen as a recursion. The term $\ln p(\mathbf{z}_{i+1})$ can be recursively replaced either with another instance of (3.5) or evaluated under the latent distribution $p(\mathbf{z}_{i+1})$, which marks the termination step. Consequently, the log density of a normalizing flow equals to:

$$\ln p(\mathbf{x}) = \ln p(\mathbf{z}_K) + \sum_{i=1}^{K} \ln |\det J_i|. \tag{3.6}$$

Normalizing flows can be conveniently sampled in two steps. First, a latent tensor $\mathbf{z}$ is sampled from the latent distribution $p(\mathbf{z})$, which is usually Gaussian. Then, the obtained tensor $\mathbf{z}$ is transformed using the inverse of function $f$ into $\mathbf{x}$. This sampling process corresponds to:

$$\mathbf{x} = f_\theta^{-1}(\mathbf{z}), \quad \mathbf{z} \sim \mathcal{N}(\mu, \Sigma). \tag{3.7}$$

This procedure corresponds to unconditional generation. Normalizing flows can also perform conditional generation, as detailed in [97]. We next describe bijections used in contemporary normalizing flow architectures [19, 95, 98, 99, 100].

## 3.2 Building blocks of the bijective flows

The standard building blocks of normalizing flows are designed to encourage efficient computation of the log density (3.6) and fast sampling. In practice, this means that the bijective building blocks have closed-form inverse and tractable computation of Jacobian determinants. All transformations process $d$-dimensional inputs $\mathbf{x} \in \mathbb{R}^d$.

**ActNorm** [99] is an invertible substitute for batch normalization [101]. It performs elementwise affine transformation with per-element scale and bias parameters:

$$\mathbf{y} = \mathbf{s} \odot \mathbf{x} + \mathbf{b}. \tag{3.8}$$

Scale $\mathbf{s}$ and bias $\mathbf{b}$ are initialized as the variance and mean of a subset of training samples, while $\odot$ stands for Hadamard (elementwise) product. Inverting the ActNorm layer is trivial, while the Jacobian is diagonal matrix, *i.e.* $\mathbf{J}_f = \mathrm{diag}(\mathbf{s})$.

**Invertible** $1 \times 1$ **Convolution** is a generalization of element permutation [99]. Convolutions with $1 \times 1$ kernel are not invertible by construction. Instead, a combination of orthogonal initialization and the loss function keeps the kernel inverse numerically stable. More specifically, the normalizing flow loss maximizes $\ln|\det \mathbf{J}_f|$ which is equivalent to maximizing $\sum_i \ln|\lambda_i|$, where $\lambda_i$ are eigenvalues of the Jacobian. Thus, maintaining a relatively large amplitude of the eigenvalues ensures a stable inversion. The Jacobian of this transformation can be efficiently computed by LU-decomposition [99].

**Affine Coupling** [98] splits the input $\mathbf{x}$ into two halves $\mathbf{x}_1 = \mathbf{x}_{1:d/2}$ and $\mathbf{x}_2 = \mathbf{x}_{d/2:d}$. The first half is propagated without changes, while the second half is affinely transformed based on the first half:

$$\mathbf{y}_1 = \mathbf{x}_1, \quad \mathbf{y}_2 = \mathbf{s} \odot \mathbf{x}_2 + \mathbf{t}, \quad (\mathbf{s}, \mathbf{t}) = \mathit{coupling\_net}(\mathbf{x}_1). \tag{3.9}$$

Parameters $\mathbf{s}$ and $\mathbf{t}$ are calculated using an arbitrary differentiable module (*coupling_net*) that is typically implemented as a residual block [98]. The two outputs $\mathbf{y}_1$ and $\mathbf{y}_2$ are concatenated to output a single tensor $\mathbf{y}$. The Jacobian of this transformation is a triangular matrix:

$$\mathbf{J}_f = \begin{bmatrix} \mathrm{I} & 0 \\ \frac{\partial \mathbf{y}_2}{\partial \mathbf{x}_1} & \mathrm{diag}(\mathbf{s}) \end{bmatrix}. \tag{3.10}$$

Thus, the determinant of the Jacobian corresponds to the product of diagonal elements. The inverse of the affine coupling layer can be done by first splitting the tensor $\mathbf{y}$ into two halves. The first half can be copied to $\mathbf{x}_1$ and used for the computation of $\mathbf{s}$ and $\mathbf{s}$. Given these values,

$\mathbf{x}_2$ can be easily computed:

$$\mathbf{x}_1 = \mathbf{y}_1, \quad \mathbf{x}_2 = (\mathbf{y}_2 - \mathbf{t})/\mathbf{s}, \quad (\mathbf{s}, \mathbf{t}) = \mathit{coupling\_net}(\mathbf{y}_1). \tag{3.11}$$

By merging ActNorm, invertible convolution and coupling layer we obtain an invertible unit (Figure 3.2) that is the essential building block of normalizing flows

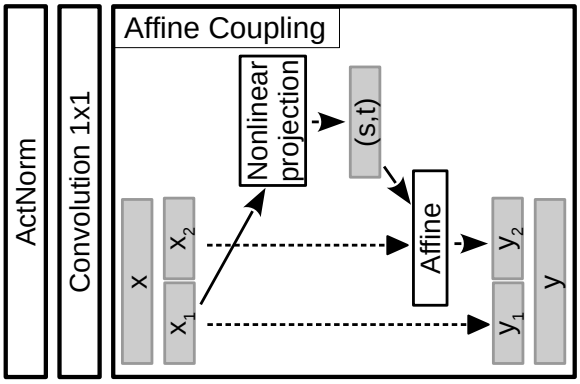


**Figure 3.2:** Example of invertible unit used in standard normalizing flow architectures.

Note that bijective functions may not have a closed form inverse, as noted in [102, 103]. However, utilizing such bijections as building blocks requires iterative inverse computation that hampers the sampling efficiency.

## 3.3 Normalizing flows for natural images

This thesis considers applying normalizing flows on natural images. Natural RGB images are usually represented as three-dimensional 8-bit tensors with discrete values from $[0, 255]$. Thus, we first transform the discrete image representations into continuous ones suitable for normalizing flows.

### 3.3.1 Dequantizing the descrete representations

Let $\mathbf{x} \in [0, 255]^{C \times H \times W}$ be a discrete image representation with $C$ channels, height $H$ and width $W$. We transform $\mathbf{x}$ into continuous representation $\mathbf{y}$ by adding uniform noise [104]:

$$\mathbf{y} = \mathbf{x} + \mathbf{u}, \quad \mathbf{u} \sim U. \tag{3.12}$$

Here, $U$ represents uniform distribution over hypercube $[0, 1[^{C \times H \times W}$. The corresponding normalizing flow takes the following form [104]:

$$\ln p_\theta(\mathbf{x}) := \ln \int p_\theta(\mathbf{x} + \mathbf{u})\, d\mathbf{u} \geq \mathbb{E}_{\mathbf{u} \sim p(\underline{\mathbf{u}})}[\ln p_\theta(\mathbf{x} + \mathbf{u})] \tag{3.13}$$

The likelihood lower bound follows from Jensen's inequality. In practice, we approximate the expectation with a single Monte Carlo sample.

The described dequantization procedure prevents the model from learning Dirac delta functions at discrete data points, which would yield unstable implementations on modern computer architectures. Still, uniform dequantization spreads the probability volume uniformly on the unit hypercube, which is often suboptimal. Thus, we can sample noise from learned posterior distribution $q(\mathbf{u}|\mathbf{x})$ that specifies noise distribution for every input image [100]. By utilizing variational inference, we can write:

$$\ln p_\theta(\mathbf{x}) := \ln \int p_\theta(\mathbf{x}+\mathbf{u}) \frac{q(\mathbf{u}|\mathbf{x})}{q(\mathbf{u}|\mathbf{x})} d\mathbf{u} \geq \mathbb{E}_{\mathbf{u}\sim q(\cdot|\mathbf{x})}[\ln p_\theta(\mathbf{x}+\mathbf{u}) - \ln q(\mathbf{u}|\mathbf{x})] \tag{3.14}$$

The distribution $q$ controls the spread of probability volume on the unit hypercube and can be parameterized with an additional set of learnable parameters [100].

The trained normalizing flow can generate discrete images by sampling the continuous representation $\mathbf{y}$ followed by quantization via floor operator $\lfloor \cdot \rfloor$:

$$\mathbf{x} = \lfloor f_\theta^{-1}(\mathbf{z}) \rfloor, \quad \mathbf{z} \sim p(\mathbf{z}). \tag{3.15}$$

### 3.3.2 Multi-scale image architecture

Mapping input images into Gaussian distribution requires sufficient capacity of nonlinear function $f_\theta$. Thus, the standard image-oriented normalizing flow architectures [98, 99, 100] stack multiple bijective layers described in Section 3.2. These transformations are applied to image tensors in a channelwise fashion.

The standard image-oriented architecture stacks multiple invertible layers that operate on a single spatial resolution into an invertible block, as visualized in Figure 3.3. Between every two consecutive steps, a portion of the latent representation is resolved according to a decoupled normal distribution while the remaining tensor is reshaped [98]. The reshape operator transforms the tensor of shape $C \times H \times W$ into $4C \times \frac{H}{2} \times \frac{W}{2}$.

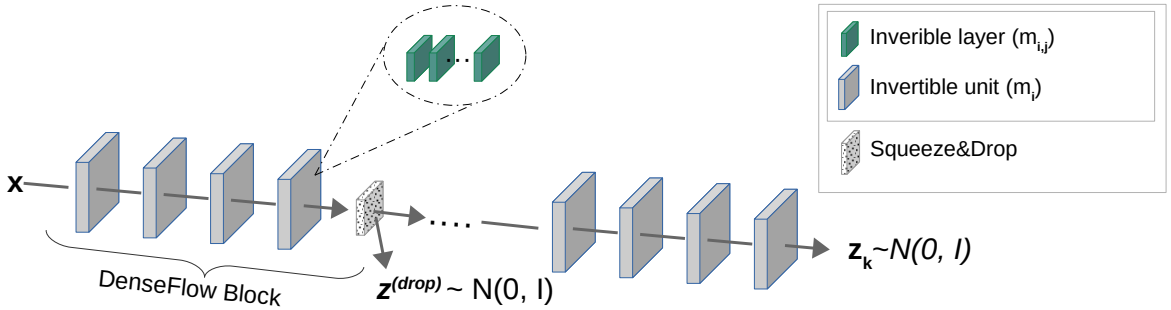


**Figure 3.3:** The standard image-oriented normalizing flow architecture.

## 3.4 Densely connected normalizing flows

Normalizing flows achieve their expressiveness by composing multiple invertible transformations [98]. This is illustrated with the scheme (3.4) where each of two consecutive latent variables $\mathbf{z}_{i-1}$ and $\mathbf{z}_i$ are connected via a dedicated flow unit $f_i$. Each flow unit $f_i$ is a bijective transformation with parameters $\theta_i$. The standard normalizing flows require that $f = f_K \circ f_{K-1} \circ \cdots \circ f_1$ be a bijective function. Contrary, we argue that the expressiveness of normalizing flows can be improved by making $f$ only piecewise bijective. This way, the dimensionality of latent representations $\mathbf{z}_i$ can be gradually enlarged. Furthermore, we introduce skip connections that can promote feature reuse [105] and smoothness of the loss landscape [106], which may explain our performance gains. Enlarging latent representations comes at the cost of exact likelihood estimation, as we show next.

### 3.4.1 Lower bound of data likelihood

Let $\mathbf{e}_i$ be a noise variable [107, 108] subjected to some known distribution $p(\mathbf{e}_i)$, e.g. a multivariate Gaussian. We can concatenate $\mathbf{e}_i$ to the intermediate latent variable $\mathbf{z}_i$ to obtain the concatenated representation $[\mathbf{z}_i, \mathbf{e}_i]$. The dimensionality of the concatenated representation $\dim(\mathbf{z}_i, \mathbf{e}_i) = \dim(\mathbf{z}_i) + \dim(\mathbf{e}_i)$. Given a noise distribution $p(\mathbf{e}_i)$ and a joint distribution $p(\mathbf{z}_i, \mathbf{e}_i)$, the log likelihood $p(\mathbf{z}_i)$ can be recovered as:

$$\ln p(\mathbf{z}_i) \geq \mathbb{E}_{\mathbf{e}_i \sim p(\mathbf{e})} \left[ \ln p(\mathbf{z}_i, \mathbf{e}_i) - \ln p(\mathbf{e}_i) \right] . \tag{3.16}$$

The detailed proof can be found in the Appendix 8.1. Thus, we enable normalizing flows to arbitrarily increase the dimensionality of the latent representations.

A tractable formulation of this idea can be obtained by estimating the expectation trough Monte Carlo sampling. In practice, we observed that a single MC sample is sufficient during the training, thus (3.16) becomes:

$$\ln p(\mathbf{z}_i) \geq \mathbb{E}_{\mathbf{e}_i \sim p(\mathbf{e})} \left[ \ln p(\mathbf{z}_i, \mathbf{e}_i) - \ln p(\mathbf{e}_i) \right] \approx \ln p(\mathbf{z}_i, \mathbf{e}_i) - \ln p(\mathbf{e}_i) . \tag{3.17}$$

Parameters of the noise distribution $p(\underline{\mathbf{e}})$ can be preselected or computed based on previous latent representations. In the latter case, we effectively introduce skip connections [105] to the normalizing flow architecture.

### 3.4.2 Skip connections through reparametrization trick

Let $\mathbf{z}_i^{(\text{aug})}$ be a random variable obtained by concatenation of intermediate $\mathbf{z}_i$ and re-parameterized noise variable $\mathbf{e}_i$, as abstracted with the function $h_i$:

$$\mathbf{z}_i^{(\text{aug})} = h_i(\mathbf{z}_i, \mathbf{e}_i; \mathbf{z}_{<i}) = [\mathbf{z}_i, \sigma \odot \mathbf{e}_i + \mu], \quad (\mu, \sigma) = g_i(\mathbf{z}_{<i}). \tag{3.18}$$

The parameters $\mu$ and $\sigma$ are computed by nonlinear transformation $g_i$ that processes previous latent representations $\mathbf{z}_{<i} = [\mathbf{z}_0, ..., \mathbf{z}_{i-1}]$. We refer to the transformation $h_i$ as *cross-unit coupling* since it acts as an affine coupling layer [95] over a group of previous invertible units. Observe that this design choice corresponds to the well-known reparametrization trick [90].

The Jacobian of $h_i$ is a diagonal square matrix:

$$\frac{\partial \mathbf{z}_i^{(\text{aug})}}{\partial [\mathbf{z}_i, \mathbf{e}_i]} = \begin{bmatrix} \mathrm{I} & 0 \\ 0 & \operatorname{diag}(\vec{\sigma}) \end{bmatrix}. \tag{3.19}$$

Here, square brackets $[\cdot, \cdot]$ denote concatenation along the features dimension. Given $\mathbf{z}_i^{(\text{aug})}$, the initial $\vec{z}_i$ can be conveniently recovered by removing the noise dimensions. This step is performed during model sampling.

We can now draw a connection between the distribution $p(\mathbf{z}_i, \mathbf{e}_i)$ from (3.16) and the distribution $p(\mathbf{z}_i^{(\text{aug})})$ as:

$$\ln p(\mathbf{z}_i, \mathbf{e}_i) = \ln p(\mathbf{z}_i^{(\text{aug})}) + \ln |\det \operatorname{diag}(\sigma)|. \tag{3.20}$$

Furthermore, we can connect $p(\mathbf{z}_i^{(\text{aug})})$ with the initial $p(\mathbf{z}_i)$ as:

$$\ln p(\mathbf{z}_i) \geq \mathbb{E}_{\mathbf{e}_i \sim p(\mathbf{e}_i)} [\ln p(\mathbf{z}_i^{(\text{aug})}) - \ln p(\mathbf{e}_i) + \ln |\det \operatorname{diag}(\sigma)|]. \tag{3.21}$$

Starting from an input variable, one can now build a normalizing flow by applying the standard transformation (3.5) or increase dimensionality through skip connections (3.21). Next, we show an example of likelihood computation with the extended normalizing flow framework.

**Example 1 (Likelihood computation)** *Let $m_1$ and $m_2$ be the bijective mappings from $\mathbf{z}_0$ to $\mathbf{z}_1$ and $\mathbf{z}_1^{(\text{aug})}$ to $\mathbf{z}_2$, respectively. Let $h_1$ be the cross-unit coupling from $\mathbf{z}_1$ to $\mathbf{z}_1^{(\text{aug})}$, $\mathbf{z}_1^{(\text{aug})} = [\mathbf{z}_1, \sigma \odot \mathbf{e}_1 + \mu]$. Assume $\sigma$ and $\mu$ are computed by any non-invertible neural network $g_1$. The network accepts $\mathbf{z}_0$ as the input. We calculate log likelihood of the input $\mathbf{z}_0$ according to the following sequence of equations: [transformation, cross-unit coupling, transformation,*

*termination].*

$$\ln p(\vec{z}_0) = \ln p(\vec{z}_1) + \ln|\det J_{f_1}|, \tag{3.22}$$

$$\ln p(\vec{z}_1) \geq \mathbb{E}_{\vec{e}_1 \sim p^*(\vec{e}_1)}[\ln p(\vec{z}_1^{(\mathrm{aug})}) - \ln p(\vec{e}_1) + \ln|\det \mathrm{diag}(\vec{\sigma})|], \quad (\vec{\sigma}, \vec{\mu}) = g_1(\vec{z}_0), \tag{3.23}$$

$$\ln p(\vec{z}_1^{(\mathrm{aug})}) = \ln p(\vec{z}_2) + \ln|\det J_{f_2}|, \tag{3.24}$$

$$\ln p(\vec{z}_2) = \ln \mathcal{N}(\vec{z}_2; 0, \mathrm{I}). \tag{3.25}$$

Note that the expectation is approximated using MC sampling with a single sample during training and a few hundred samples during evaluation. Still, our extended framework generates samples with a single pass since the inverse does not require MC sampling nor utilization of $g_1$.

Figure 3.4 compares the standard normalizing flow (a) normalizing flow with input augmentation [107] (b) and the proposed densely connected incremental augmentation with cross-unit coupling (c). Each flow unit $f_i^{\mathrm{DF}}$ consists of several invertible modules $m_{i,j}$ and cross-unit coupling $h_i$. The main novelty of our architecture is that each flow unit $f_{i+1}^{\mathrm{DF}}$ increases the dimensionality with respect to its predecessor $f_i^{\mathrm{DF}}$. Cross-unit coupling $h_i$ augments the latent variable $\mathbf{z}_i$ with affinely transformed noise $\mathbf{e}_i$. Parameters of the affine noise transformation are obtained by a nonlinear function $g_i$ which accepts all previous variables $\mathbf{z}_{<i}$.

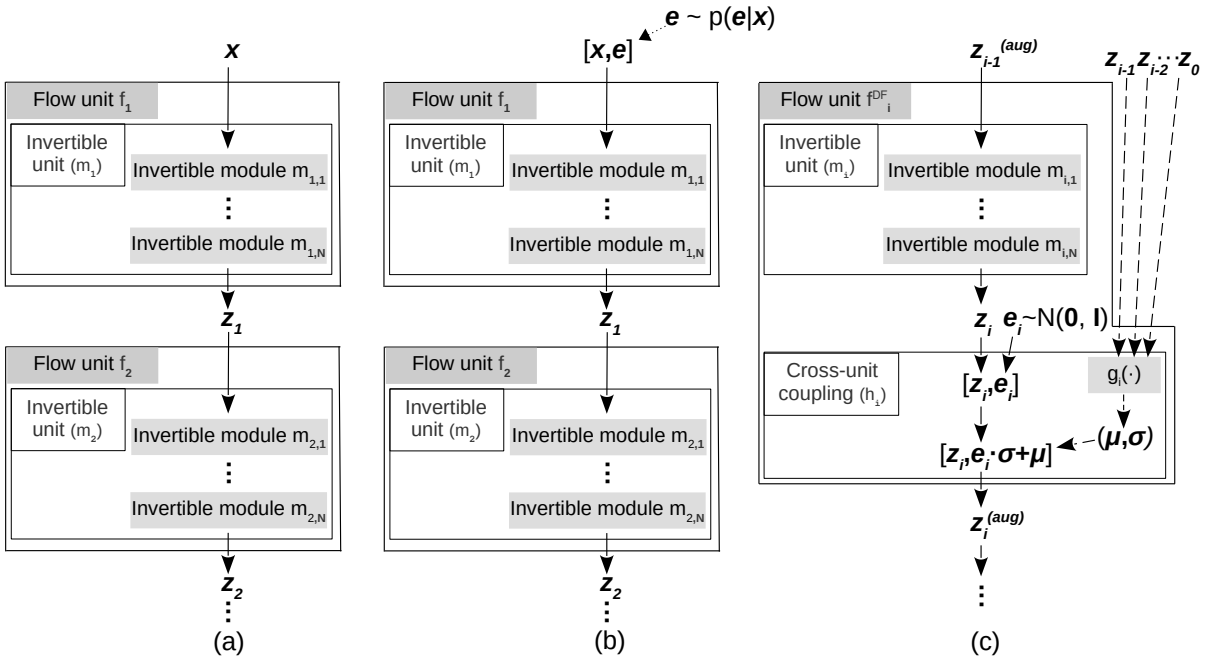


**Figure 3.4:** Standard normalizing flow [95, 98] (a), normalizing flow with augmented input [107] (b), and the proposed incremental augmentation with cross-unit coupling (c). Unlike (b) which adds noise only to the input, (c) adds noise to the output of every unit except the last.

We repeatedly apply the cross-unit coupling $h_i$ throughout the architecture to achieve incremental augmentation of intermediate latent representations. Consequently, the data distribution

is modeled in a latent space of higher dimensionality than the input space [107, 108].

### 3.4.3 *DenseFlow*: densely connected image architecture

We construct *DenseFlow*, an image-oriented architecture that extends multi-scale Glow [99] with incremental augmentations of latent representations through cross-unit coupling. Each DenseFlow block consists of several DenseFlow units and resolves a portion of the latent representation according to a decoupled normal distribution [98]. Each DenseFlow unit $f_i^{\mathrm{DF}}$ consists of $N$ invertible layers ($m_i = m_{i,N} \circ \cdots \circ m_{i,1}$) and cross-unit coupling ($h_i$). Note that our glow-like modules build coupling networks with densely connected blocks [105] and Nyström self-attention [109], as visualized in Figure 3.5. On the contrary, the standard glow-like modules [99] rely on residual blocks [110].

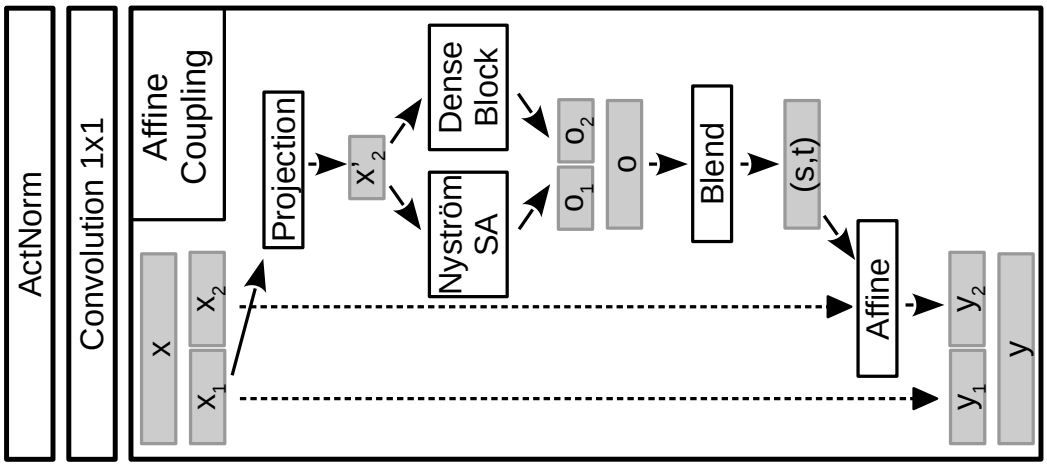


**Figure 3.5:** Our affine coupling implements the coupling network as a combination of densely connected blocks [105] and Nyström self-attention [109].

The input to each DenseFlow unit is the output of the previous unit augmented with the noise and transformed in the cross-unit coupling fashion. The number of introduced noise channels is defined as the growth-rate hyperparameter. Generally, the number of invertible modules in latter DenseFlow units should increase due to enlarged latent representation. We stack $M$ DenseFlow units to form a DenseFlow block. The last invertible unit in the block does not have the corresponding cross-unit coupling. We stack multiple DenseFlow blocks to form a normalizing flow with a large capacity. Between each two blocks, we decrease the spatial resolution and compress the latent representation by introducing a squeeze-and-drop module [98]. The squeeze-and-drop module applies space-to-channel reshaping and resolves half of the dimensions according to the prior distribution. We denote the developed architecture as *DenseFlow-L-k*, where $L$ is the total number of invertible modules while $k$ denotes the growth rate. The developed architecture uses two independent levels of skip connections. The first level (intra-module) is formed of skip connections inside every coupling network. The second level

(cross-unit) connects DenseFlow units at the top level of the architecture.

Figure 3.6 shows the final DenseFlow architecture. Grey squares represent DenseFlow units. Cross-unit coupling is represented with blue dots and dashed skip connections. Finally, squeeze-and-drop operations between successive DenseFlow blocks are represented by dotted squares. The proposed DenseFlow design applies invertible but less powerful transformations (e.g. convolution $1 \times 1$) on tensors of larger dimensionality. On the other hand, powerful non-invertible transformations such as coupling networks perform most of their operations on lower-dimensional tensors. This leads to resource-efficient training and inference.

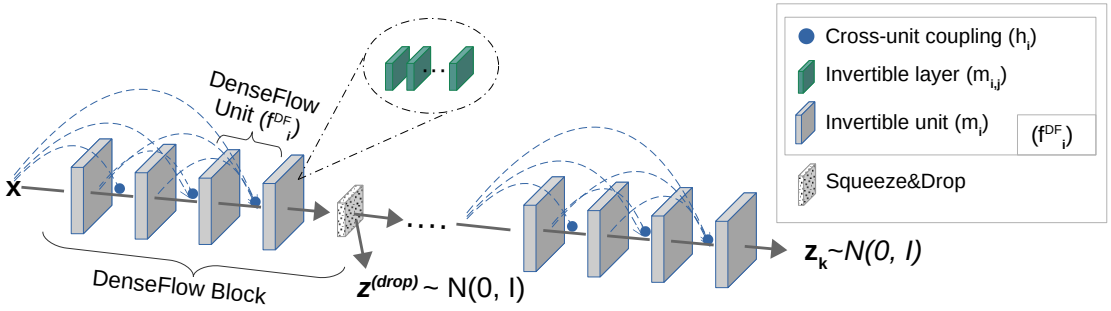


**Figure 3.6:** The proposed DenseFlow architecture. DenseFlow blocks consist of DenseFlow units ($f_i^{\mathrm{DF}}$) and a Squeeze-and-Drop module [98]. DenseFlow units are densely connected through cross-unit coupling ($h_i$). Each DenseFlow unit includes multiple invertible modules ($m_{i,j}$) from Figure 3.5.

We use DenseFlow architecture to model synthetic negative samples required for the training of our open-set segmentation models, as we describe in the following chapter.

# Chapter 4

# Hybrid open-set segmentation of images

This chapter constructs open-set segmentation by overriding closed-set predictions with dense anomaly detection, as presented in Section 4.1. Section 4.2 formulates a novel anomaly score as a hybrid ensemble of generative and discriminative cues that build upon shared dense semantic features. Our generative score corresponds to unnormalized joint density that is very hard to train due to intractable integral in the normalization constant. Section 4.3 proposes an elegant solution that involves training on mixed-content images with pasted negative content. Finally, we relax the requirement for a negative training dataset by synthesizing the negative content with a jointly trained normalizing flow, as presented in Section 4.4.

**Notation.** We extend the notation from Chapter 3 and define pixel label at the location $(i, j)$ as a categoric random variable $\underline{y}^{ij}$ that takes values from a set $\mathscr{Y}$ of size $K = |\mathscr{Y}|$. A binary random variable $\underline{d}^{ij}$ models whether the given pixel is an inlier or an outlier. The realization of a random variable is denoted by omitting the underline while the spatial locations are often omitted for brevity. Thus, $P(y|\mathbf{x})$ and $P(d|\mathbf{x})$ are shortcuts for $P(\underline{y}^{ij} = y^{ij}|\underline{\mathbf{x}} = \mathbf{x})$ and $P(\underline{d}^{ij} = d^{ij}_{\text{in}}|\underline{\mathbf{x}} = \mathbf{x})$, where we write $d^{ij}_{\text{in}}$ if a pixel at location $(i, j)$ is inlier and $d^{ij}_{\text{out}}$ if the pixel is outlier.

## 4.1 Open-set segmentation

Open-set segmentation simultaneously recognizes the known classes and identifies anomalous pixels. We formulate open-set recognition by complementing a pre-trained classifier with semantic anomaly detection. In particular, we override the closed-set segmentation output in pixels detected as semantic anomalies. This procedure enables simultaneous segmentation of known classes and identification of anomalous parts of the scene as presented in Figure 4.1. The described approach relies on accurate anomaly detection to attain satisfactory open-set predictions. Thus, we develop an accurate dense semantic anomaly detector that fuses discriminative and generative cues into a hybrid anomaly score as we describe next. Many real-world ap-

plications necessitate real-time inference. Thus, we design an anomaly detector that does not significantly increase the computational cost.

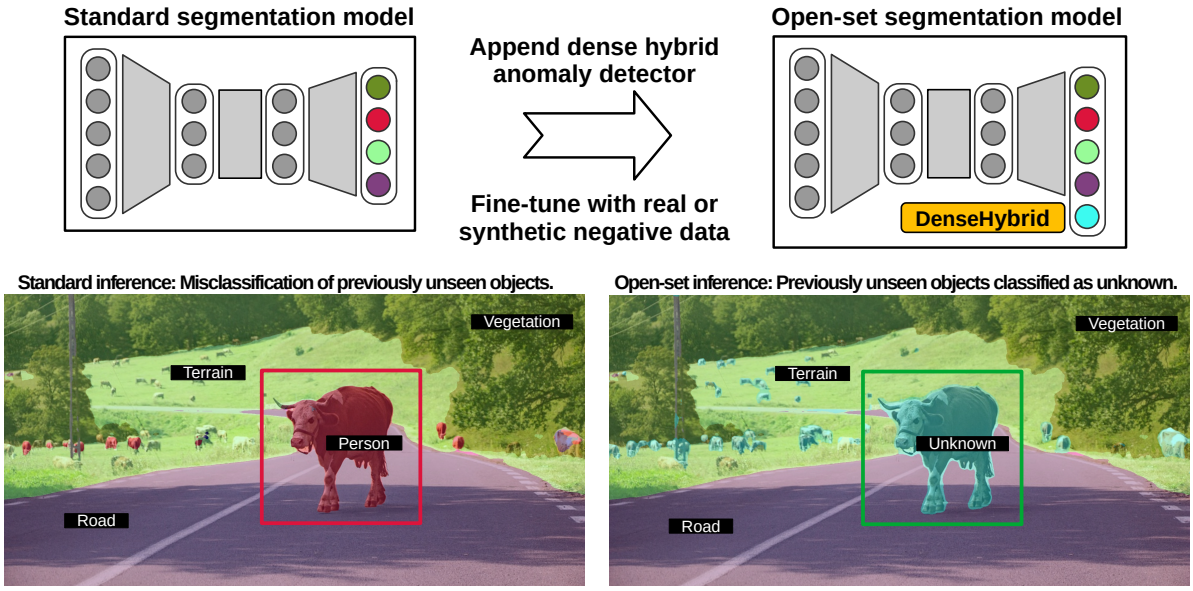


**Figure 4.1:** Dense open-set inference with the standard segmentation models can be achieved by appending dense anomaly detector that overrides decision in anomalous pixels.

## 4.2 Hybrid anomaly detection

Many existing methods for anomaly detection can be categorized as generative and discriminative approaches. Typical generative approaches estimate the likelihood [13, 32] or resynthesise the input [50, 51], while discriminative approaches model the decision boundary between inliers and outliers [10, 12]. We observe that the two approaches exhibit different failure modes and thus can be joined into a hybrid anomaly score. To show this, we consider the following toy example. Figure 4.2 presents three anomaly detection approaches on a two-dimensional toy problem (details in Appendix 8.2). The discriminative approach models the inlier posterior $P(d_{\text{in}}|\mathbf{x})$. It often fails far from the inliers since a finite negative training dataset cannot cover all modes of the test anomalies. The generative approach models the data likelihood $p(\mathbf{x})$. It often errs along the boundary of the inlier manifold due to over-generalization [41, 48], but does not expand into the open space. We ensemble these two approaches since they tend to assume different failure modes. Hybrid anomaly score alleviates both the coarseness of the generative approach and the inaccuracy of the discriminative approach far from the training negatives. This synergy favors accurate boundaries near the negative training data while reducing false negative anomalies in the open space.

Following the described intuition we now state a sufficient condition for the performance gain of our hybrid ensemble over each of its two components. Let $s : \mathcal{X} \to \mathbb{R}$ be a standardized

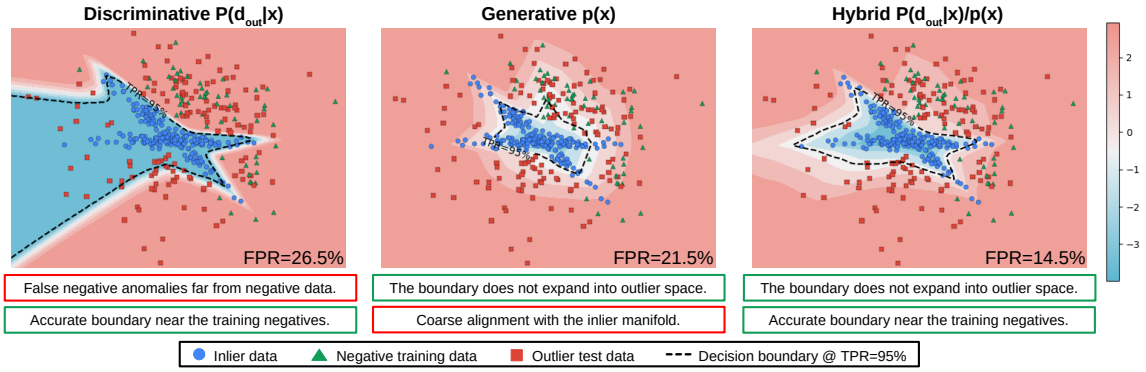


**Figure 4.2:** Three anomaly detection approaches on a toy problem. Inliers, train negatives and test anomalies are shown as blue, green and red points (details in the Appendix). The background heatmaps designate the three anomaly scores with higher values in red. The discriminative anomaly score (left) is susceptible to false negative responses since the negative training dataset is finite and cannot cover all modes of test anomalies. The generative anomaly score (middle) errs along the border of the inlier manifold due to over-generalization [41, 48], but is unlikely to commit errors far from the inlier manifold. Our hybrid approach prevails by ensembling discriminative and generative cues.

anomaly score which assigns higher values to anomalies. We can decompose the score $s$ into correct labeling $f$ and error $\varepsilon$:

$$s(\mathbf{x}) = f(\mathbf{x}) + \varepsilon(\mathbf{x}). \tag{4.1}$$

Function $f : \mathscr{X} \to \{-1, +1\}$ labels anomalies with +1 and inliers with -1. The expected squared error then equals:

$$\mathscr{E}(s) = \mathbb{E}_{\mathbf{x}}[(s(\mathbf{x}) - f(\mathbf{x}))^2] = \mathbb{E}_{\mathbf{x}}[(\varepsilon(\mathbf{x}))^2]. \tag{4.2}$$

Our goal is to show conditions under which the hybrid anomaly score outperforms both of its components:

$$\mathscr{E}(s_H) < \inf\{\mathscr{E}(s_G), \mathscr{E}(s_D)\}. \tag{4.3}$$

The generative anomaly score $s_G$ is a function of data likelihood. The discriminative anomaly score $s_D$ is a function of inlier posterior. By defining our hybrid anomaly detector as $s_H(\mathbf{x}) := \frac{1}{2}s_D(\mathbf{x}) + \frac{1}{2}s_G(\mathbf{x})$, the condition (4.3) becomes as follows (proof in the Appendix 8.3):

$$\frac{\alpha - 3}{4} e + C_1 \rho(\varepsilon_D, \varepsilon_G) + C_2 < 0 \tag{4.4}$$

Here $\rho$ is the Pearson correlation coefficient between the errors, $\alpha = \frac{\sup\{\mathscr{E}(s_G), \mathscr{E}(s_D)\}}{\inf\{\mathscr{E}(s_G), \mathscr{E}(s_D)\}}$ denotes the error ratio of the two components, $e = \inf\{\mathscr{E}(s_G), \mathscr{E}(s_D)\}$ denotes the smallest expected error, while $C_1$ and $C_2$ can be viewed as constants. If the errors of the two components are independent and Gaussian ($\rho = 0, C_1 = 0.5$ and $C_2 = 0$), then our hybrid anomaly detector will be effective even if $\alpha < 3$. The condition (4.4) can be satisfied even when the two components are moderately correlated as in our experiments. This creates an opportunity to build efficient hybrid anomaly detectors atop generative and discriminative detectors with shared features.

### 4.2.1 Efficient implementation atop semantic classifier

Standard semantic segmentation can be viewed as a two-step procedure. Given an input image $\mathbf{x}$, a deep feature extractor $f_{\theta_1}$ computes an abstract representation $\mathbf{z}$ also known as pre-logits. Then, the computed pre-logits are projected into logits $\mathbf{s}$ and activated by softmax. The softmax output models the class posterior $P(y|\mathbf{x})$:

$$P(y|\mathbf{x}) := \text{softmax}(\mathbf{s}_y), \text{ where } \mathbf{s} = f_{\theta_2}(\mathbf{z}), \mathbf{z} = f_{\theta_1}(\mathbf{x}). \tag{4.5}$$

In practice, $f_{\theta_1}$ can be any dense feature extractor that is suitable for semantic segmentation, while $f_{\theta_2}$ is a simple projection. We extend this framework with dense data likelihood and discriminative inlier posterior, the two components of our hybrid anomaly score.

**Dense data likelihood.** Dense data likelihood can be expressed atop the dense classifier $f_{\theta_2}$ by re-interpreting exponentiated logits $\mathbf{s}$ as unnormalized joint density [17]:

$$\hat{p}(y,\mathbf{x}) := \exp(\mathbf{s}_y), \quad \mathbf{s} = f_{\theta_2}(f_{\theta_1}(\mathbf{x}))\,. \tag{4.6}$$

We can now recover dense data likelihood trough marginalization of $\underline{y}$:

$$p(\mathbf{x}) = \sum_y p(y,\mathbf{x}) = \frac{1}{Z}\sum_y \hat{p}(y,\mathbf{x}) = \frac{1}{Z}\sum_y \exp \mathbf{s}_y. \tag{4.7}$$

Here, the unnormalized likelihood corresponds to $\hat{p}(\mathbf{x}) = \sum_y \exp \mathbf{s}_y$ and $Z$ denotes the normalization constant dependent only on model parameters. As usual, $Z$ is intractable since it requires aggregating the unnormalized distribution for all realizations of $\underline{y}$ and $\underline{\mathbf{x}}$:

$$Z = \int_{\mathbf{x}} \sum_y \exp \mathbf{s}_y. \tag{4.8}$$

Throughout this work, we conveniently eschew the evaluation of $Z$ in order to enable efficient training and inference.

**Class posterior.** The standard discriminative predictions (4.5) can still be consistently recovered according to Bayes rule [17]:

$$P(y|\mathbf{x}) = \frac{p(y,\mathbf{x})}{p(\mathbf{x})} = \frac{p(y,\mathbf{x})}{\sum_{y'} p(y',\mathbf{x})} = \frac{\frac{1}{Z}\hat{p}(y,\mathbf{x})}{\sum_{y'} \frac{1}{Z}\hat{p}(y',\mathbf{x})} = \frac{\exp \mathbf{s}_y}{\sum_{y'} \exp \mathbf{s}_{y'}} = \text{softmax}(\mathbf{s}_y). \tag{4.9}$$

The normalization constant $Z$ appears both in the numerator and denominator and hence cancels out. Reinterpretation of logits (4.7) enables convenient unnormalized per-pixel likelihood estimation atop pre-trained dense classifiers. Note that adding a constant value to the logits does not affect the standard classification but affects our formulation of data likelihood. We exploit the

extra degree of freedom to formulate the generative anomaly score $s_G(\mathbf{x}) \propto -\ln \hat{p}(\mathbf{x})$. The same extra degree of freedom has been used to model a discriminator network in semi-supervised learning [111].

**Inlier posterior.** We define the inlier posterior $P(d_{\text{in}}|\mathbf{x})$ as a non-linear transformation $g_\gamma$ of pre-logits $\mathbf{z}$ [10]:

$$P(d_{\text{in}}|\mathbf{x}) = 1 - P(d_{\text{out}}|\mathbf{x}) := \sigma(g_\gamma(\mathbf{z})). \tag{4.10}$$

Here, function $g_\gamma$ is an additional projection parameterized with $\gamma$. Thus the initial set of parameters $\theta$ is extended with $\gamma$. The discriminative anomaly score $s_D(\mathbf{x}) \propto \ln P(d_{\text{out}}|\mathbf{x})$.

**Hybrid anomaly score.** Finally, we materialize our hybrid anomaly score as a likelihood ratio that can also be interpreted as ensemble $s_H(\mathbf{x}) = s_D(\mathbf{x}) + s_G(\mathbf{x})$:

$$s_H(\mathbf{x}) := \ln \frac{P(d_{\text{out}}|\mathbf{x})}{p(\mathbf{x})} \cong \ln P(d_{\text{out}}|\mathbf{x}) - \ln \hat{p}(\mathbf{x}). \tag{4.11}$$

We refer to this hybrid anomaly score as *DenseHybrid*. Our generative component can neglect $Z$ since the ranking performance [24] is invariant to monotonic transformation such as taking a logarithm or adding a constant. The detailed derivation and connection with the ensemble of the two components is in the Appendix. This particular formulation equalizes the influence of the two components. Still, other definitions may also be effective, which is an interesting direction for future work. Note that DenseHybrid is remarkably well suited for dense prediction due to minimal overhead and translational equivariance.

### 4.2.2 Dense open-set inference

The proposed hybrid anomaly detector can be combined with the closed-set output to recover open-set predictions as shown in Figure 4.3. The input image is fed to a dense feature extractor which produces pre-logits $\mathbf{z}$ and logits $\mathbf{s}$. We recover the closed-set posterior $P(y|\mathbf{x})$ with softmax, and the unnormalized data log-likelihood $\ln \hat{p}(\mathbf{x})$ with log-sum-exp (designated in green). A distinct head $g$ transforms pre-logits $\mathbf{z}$ into the inlier posterior $P(d_{\text{in}}|\mathbf{x})$ (designated in yellow). The anomaly score $s(\mathbf{x})$ is a log ratio between dataset-posterior and density (4.11). The resulting anomaly map is thresholded and fused with the discriminative output into the final dense open-set output. The anomaly threshold is usually selected on a validation set that consists of inliers and outliers. The desired behaviour of the dense hybrid open-set model is attained by fine-tuning a pre-trained classifier, as we describe next.

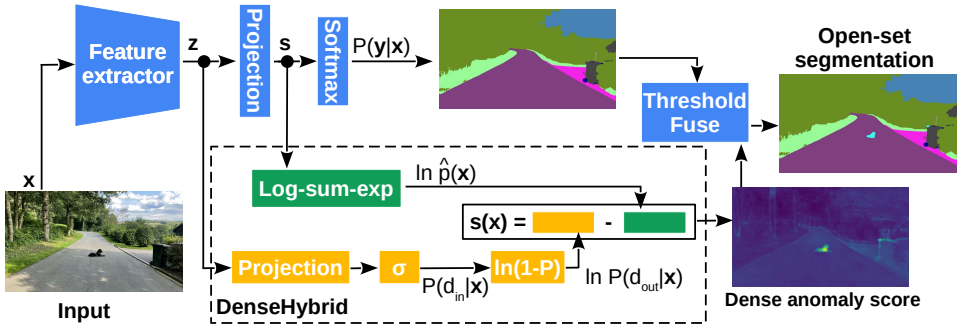


**Figure 4.3:** Our open-set segmentation approach complements any semantic segmentation model which recovers dense logits with our hybrid anomaly detection. Our dense anomaly score is a log-ratio of inlier posterior and data likelihood. We implement open-set segmentation by overriding the closed-set output with thresholded anomaly score.

## 4.3 Open-set training with real negative data

Our open-set approach complements an arbitrary closed-set segmentation model with the DenseHybrid anomaly detector. Our hybrid open-set model requires joint fine-tuning of three dense prediction heads: closed-set class posterior $P(y|\mathbf{x})$, unnormalized data likelihood $\hat{p}(\mathbf{x})$ [17], and inlier posterior $P(d_{\text{in}}|\mathbf{x})$ [10]. We propose a novel training setup that eschews the intractable normalization constant by introducing negative data to the generative learning objective. The same negative data is used to train the inlier posterior. The corresponding training objectives are presented in the following paragraphs.

**Class posterior.** The closed-set class-posterior head can be trained according to the standard discriminative cross-entropy loss over the inlier dataset $D_{\text{in}}$:

$$\begin{aligned} L_{\text{cls}}(\theta) &= \mathbb{E}_{\mathbf{x},y\in D_{\text{in}}}[-\ln P(y|\mathbf{x})] \\ &= \mathbb{E}_{\mathbf{x},y\in D_{\text{in}}}[-\mathbf{s}_y] + \mathbb{E}_{\mathbf{x},y\in D_{\text{in}}}[\operatorname*{LSE}_{y'}(\mathbf{s}_{y'})]. \end{aligned} \tag{4.12}$$

As before, $\mathbf{s}$ are logits computed by $f_\theta$, while LSE stands for log-sum-exp where the sum iterates over classes.

**Data likelihood.** Training unnormalized likelihood can be a daunting task since backpropagation through $p(\mathbf{x})$ involves intractable integration over all possible images [87, 112]. Previous MCMC-based solutions [17] are not feasible in our setup due to high-resolution inputs and dense prediction. We eschew the normalization constant by optimizing the likelihood both in inlier and outlier pixels:

$$\begin{aligned} L_{\mathbf{x}}(\theta) &= \mathbb{E}_{\mathbf{x}\in D_{\text{in}}}[-\ln p(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\in D_{\text{out}}}[-\ln p(\mathbf{x})] \\ &= \mathbb{E}_{\mathbf{x}\in D_{\text{in}}}[-\ln \hat{p}(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\in D_{\text{out}}}[-\ln \hat{p}(\mathbf{x})] \end{aligned} \tag{4.13}$$

Note that the normalization constant $Z$ cancels out due to training with outliers, as detailed in

the Appendix 8.4. In practice, we use a simplified loss that is an upper bound of the above expression ($L_{\mathbf{x}}^{\mathrm{UB}} \geq L_{\mathbf{x}}$):

$$L_{\mathbf{x}}^{\mathrm{UB}}(\theta) = \mathbb{E}_{\mathbf{x},y\in D_{\mathrm{in}}}[-\mathbf{s}_y] + \mathbb{E}_{\mathbf{x}\in D_{\mathrm{out}}}[\operatorname*{LSE}_{y'}(\mathbf{s}_{y'})]. \tag{4.14}$$

We observe that $L_{\mathrm{cls}}$ and $L_{\mathbf{x}}^{\mathrm{UB}}$ have a shared loss term. Recall that training data likelihood only on inliers [17, 87] would require MCMC sampling, which is infeasible in our context. Unnormalized likelihood could also be trained through score matching [112]. However, this would preclude hybrid modelling due to having to train on noisy inputs. Consequently, it appears that the proposed training approach is a method of choice in our context. Comparison of the discriminative loss (4.12) and the generative upper bound (4.14) reveals that the standard classification loss is well aligned with the upper bound in inlier pixels. The proof of inequality $L_{\mathbf{x}}^{\mathrm{UB}} \geq L_{\mathbf{x}}$ is in the Appendix 8.4.

**Inlier posterior.** The dataset-posterior head $P(d_{\mathrm{in}}|\mathbf{x})$ requires a discriminative loss that distinguishes the inliers $\mathbf{x} \in D_{\mathrm{in}}$ from the outliers $\mathbf{x} \in D_{\mathrm{out}}$ [10]:

$$\begin{aligned} L_{\mathbf{d}}(\theta,\gamma) = &- \mathbb{E}_{\mathbf{x}\in D_{\mathrm{in}}}[\ln P(d_{\mathrm{in}}|\mathbf{x})] \\ &- \mathbb{E}_{\mathbf{x}\in D_{\mathrm{out}}}[\ln(1-P(d_{\mathrm{in}}|\mathbf{x}))]. \end{aligned} \tag{4.15}$$

**Compound loss.** Our final compound loss aggregates $L_{\mathrm{cls}}$, $L_{\mathbf{x}}^{\mathrm{UB}}$ and $L_{\mathbf{d}}$:

$$\begin{aligned} L(\theta,\gamma) = &- \mathbb{E}_{\mathbf{x},y\in D_{\mathrm{in}}}[\ln P(y|\mathbf{x}) + \ln P(d_{\mathrm{in}}|\mathbf{x})] \\ &- \mathbb{E}_{\mathbf{x}\in D_{\mathrm{out}}}[\ln(1-P(d_{\mathrm{in}}|\mathbf{x})) - \ln \hat{p}(\mathbf{x})]. \end{aligned} \tag{4.16}$$

In practice, we use a modulation hyperparameter for every loss component. More on the hyperparameters, together with the complete derivation, can be found in the Appendix 8.5.

Figure 4.4 illustrates the training of our open-set segmentation models. The figure shows that we prepare mixed-content training images $\mathbf{x}'$ by pasting negative patches $\mathbf{x}^- \in D_{\mathrm{out}}$ into regular training images $\mathbf{x}^+ \in D_{\mathrm{in}}$:

$$\mathbf{x}' = (\mathbf{1}-\mathbf{m}) \cdot \mathbf{x}^+ + \mathrm{pad}(\mathbf{x}^-, \mathbf{m}). \tag{4.17}$$

The binary mask $\mathbf{m}$ identifies negative pixels within the mixed-content image $\mathbf{x}'$. Semantic labels of negative pixels are set to void. The resulting mixed-content image $\mathbf{x}'$ is fed to the segmentation model that produces pre-logits $\mathbf{z}$ and logits $\mathbf{s}$. We recover the class posterior, unnormalized likelihood, and inlier posterior as explained in Sec. 4.2.1, and perform the training with respect to the loss (4.16).

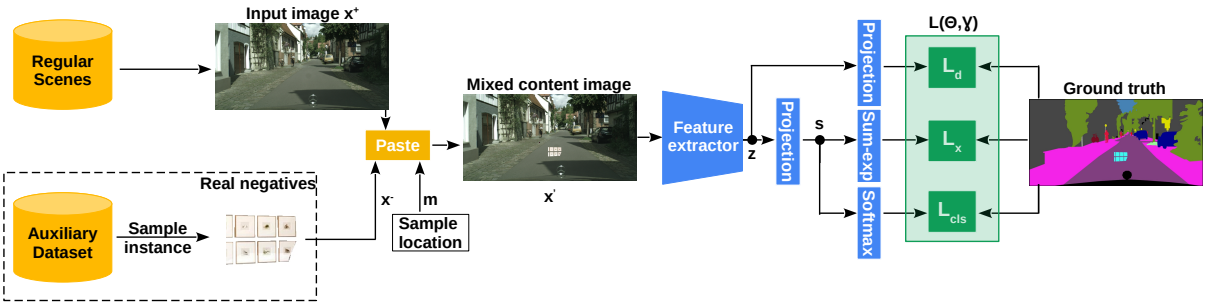


**Figure 4.4:** Fine-tuning procedure for the proposed open-set model with the DenseHybrid anomaly detector. Mixed-content images are constructed by pasting negatives sourced from an auxiliary real dataset into inlier images according to (4.17). Mixed-content images are then fed to the open-set model that produces three dense outputs: the closed-set class posterior, unnormalized data likelihood, and inlier posterior. The model is optimized according to the compound loss (4.16).

## 4.4 Open-set training with synthetic negative data

Training anomaly detectors on real negative training data may result in over-optimistic performance estimates due to a non-empty intersection between the training negatives and test anomalies. An exciting approach to relieve the dependency on real negative data is to replace them with samples from a suitably trained generative model [55, 71, 72, 73]. In such a case, the generative model is trained to generate synthetic samples that encompass the inlier distribution [71]. The required learning signal can be derived from discriminative predictions [55, 71, 73] or provided by an adversarial module [76]. Anyway, replacing real negative data with synthetic counterparts requires joint training of the generative model. We choose a normalizing flow [19] due to fast training, good distributional coverage, and fast generation at varying spatial dimensions [74]. Normalizing flows are elaborated in the previous chapter.

We train the normalizing flow $p_\zeta$ according to the data term and boundary-attraction term. The data term $L_{\text{mle}}$ corresponds to image-wide negative log-likelihood of random crops from inlier images $\mathbf{x}^+$:

$$L_{\text{mle}}(\zeta) = -\mathbb{E}_{\mathbf{x}^+ \in D_{\text{in}}}[\ln p_\zeta(\text{crop}(\mathbf{x}^+, \mathbf{m}))]. \tag{4.18}$$

The crop notation mirrors the pad notation from (4.17). Random crops vary in spatial resolution. This term aligns the generative distribution with the distribution of the training data. It encourages coverage of the inlier distribution assuming sufficient capacity of the generative model.

The boundary-attraction term $L_{\text{jsd}}$ [74] corresponds to negative Jensen-Shannon divergence between the class-posterior and the uniform distribution across all generated pixels:

$$L_{\text{jsd}}(\zeta;\theta) = \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,\mathbf{I})}[\text{JSD}[U||p_\theta(y|h_\zeta^{-1}(\mathbf{z}))]]. \tag{4.19}$$

This term pushes the generative distribution towards the periphery of the inlier distribution where the class posterior should have a high entropy. Note that gradients of this term must propagate through the entire segmentation model in order to reach the normalizing flow. Hence, the flow is penalized when the generated sample yields high softmax confidence. This signal pushes the generative distribution away from high-density regions of the input space [71]. The total normalizing flow loss modulates the contribution of the boundary term with hyperparameter $\lambda$:

$$L(\zeta;\theta) = L_{\text{mle}}(\zeta) + \lambda \cdot L_{\text{jsd}}(\zeta;\theta) \tag{4.20}$$

Optimization of (4.20) enforces the generative distribution to encompass the inlier distribution. Note that our normalizing flow can never match the diversity of images from a real dataset such as ADE20k. It would be unreasonable to expect a generation of a sofa after training on traffic scenes. Still, if the flow succeeds to learn the boundary of the inlier distribution, then DenseHybrid will be inclined to associate all off-distribution datapoints with low $s_H$.

Joint training of synthetic negatives and hybrid open-set model follows an alternating optimization procedure. In the first step, the open-set classifier is updated using synthetic negatives generated by the current parameters of the generative model $\zeta$. In the second step, the generative model is refined based on the open-set model $\theta$ from the previous step. These two steps are repeated alternately for a predetermined number of iterations.

Figure 4.5 details the training procedure with synthetic negatives. We sample the normalizing flow by *i)* selecting a random spatial resolution $(H_o, W_o)$ from a predefined interval, *ii)* sampling a random latent representation $\mathbf{z}_o \sim \mathcal{N}(0, \mathrm{I}_{H_o W_o})$, and *iii)* feeding $\mathbf{z}_o$ to the flow so that $\mathbf{x}^- = h_\zeta^{-1}(\mathbf{z}_o)$. We again craft a mixed-content image $\mathbf{x}'$ by pasting the synthesized negative patch $\mathbf{x}^- \sim p_\zeta$ into the regular training image $\mathbf{x}^+ \in D_{\text{in}}$ according to (4.17), perform the forward pass, determine $L_{\text{cls}}$, $L_{\mathbf{d}}$, $L_{\mathbf{x}}$, and $L_{\text{jsd}}$, and recover the training gradients by backpropagation. We now take the deleted inlier patch $\mathbf{x}_\mathbf{s}^+$, perform inference with the normalizing flow ($\mathbf{z}_o = h_\zeta(\mathbf{x}_\mathbf{s}^+)$) and accumulate gradients of $L_{\text{mle}}$ before performing a model-wide parameter update.

We can also source the negative content from a mixture of real and synthetic samples, as detailed on Figure 4.6. Then, the amount of data from each source is modulated by hyperparameter $b \in [0,1]$. The probability of sampling a real negative equals $b$, while the probability of sampling a synthetic negative equals $1-b$. Hence, the distribution of mixed negatives $p_{\text{neg}}$ is:

$$p_{\text{neg}}(\mathbf{x}^-) = b \cdot p_{\text{out}}(\mathbf{x}^-) + (1-b) \cdot p_\zeta(\mathbf{x}^-),\ b \in [0,1] \tag{4.21}$$

Sampling $p_{\text{neg}}$ proceeds by first choosing the source, which corresponds to sampling a Bernoulli distribution $\mathcal{B}(b)$. Then, the negative is generated by sampling the selected source. Note that we only require a set of data points ($D_{\text{out}}$) collected from $p_{\text{out}}$ without a closed-form definition of $p_{\text{out}}$. While training on mixed negative data is possible, the experimental evaluation did not

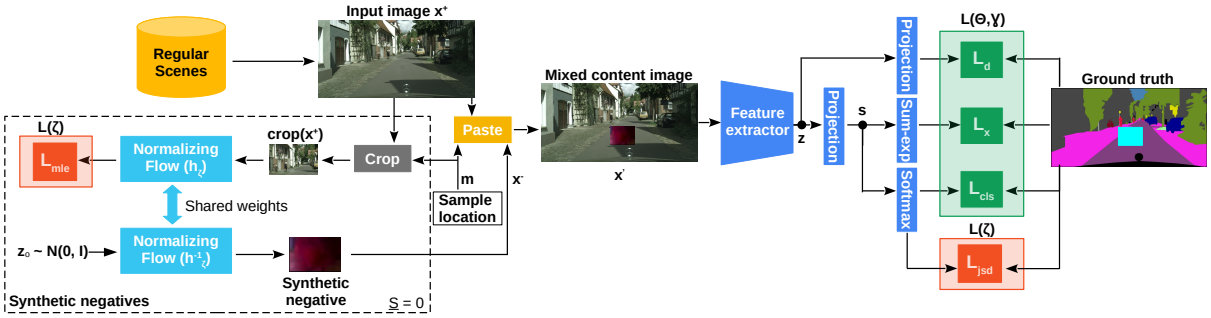


**Figure 4.5:** Fine-tuning procedure for the proposed open-set model with the DenseHybrid anomaly detector. Mixed-content images are constructed by pasting negatives sampled from a normalizing flow into inlier images. The mixed-content images are then fed to the open-set model that produces three dense outputs: the closed-set class posterior, unnormalized data likelihood, and inlier posterior. The normalizing flow maximizes the likelihood of inlier crops while aiming to generate patches that yield high entropy predictions. The dense classifier and normalizing flow are then jointly learned by optimizing (4.16) and (4.20) respectively.

reveal performance gains over exclusive training on real negative data. We next evaluate the proposed DenseHybrid in semantic anomaly detection and subsequently in open-set segmentation.

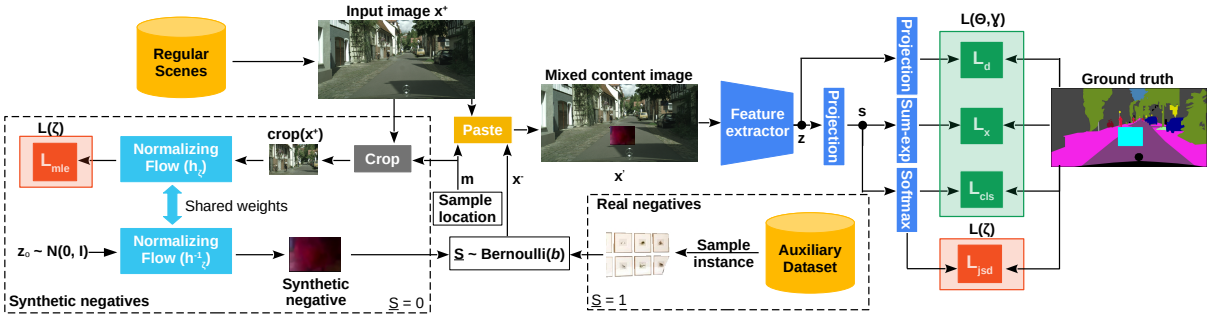


**Figure 4.6:** Fine-tuning procedure for the proposed DenseHybrid model. We construct mixed-content images by pasting negatives into inlier images according to (4.17). The negative training data can be sourced from an auxiliary real dataset, from a jointly trained normalizing flow, or from both sources according to $b$ from (4.21). Mixed-content images are fed to the open-set model that produces three dense outputs: the closed-set class posterior, unnormalized data likelihood, and inlier posterior. The model is optimized according to the compound loss (4.16). In the case of synthetic negatives ($S = 0$), the normalizing flow optimizes the loss (4.20).

### 4.4.1 Coverage-oriented generation of synthetic negatives

Prior works [71, 72] rely on GANs for generating synthetic negatives. We argue that normalizing flows offer better coverage of inlier distribution [48] and thus yield more diverse negative samples. Our argument proceeds by analyzing the gradient of the loss (4.20) with respect to the generator of synthetic negatives for both approaches. For brevity, we omit loss modulation hyperparameters nad consider simpler image-wide case.

Adversarial outlier-aware learning [71] jointly optimizes the zero-sum game between the generator $G_\psi$ and the discriminator $D_\phi$, closed-set classification $P_\theta$, and the confidence objective that enforces uncertain classification in the negative data points [71]:

$$L_{\text{adv}}(\phi,\psi;\theta) = \int p_D(\mathbf{x}) \ln D_\phi(\mathbf{x})\, d\mathbf{x} + \int p_{G_\psi}(\mathbf{x}) \ln(1 - D_\phi(\mathbf{x}))\, d\mathbf{x} \\ - \int \sum_y p_D(y,\mathbf{x}) \ln P_\theta(y|\mathbf{x})\, d\mathbf{x} + \int p_{G_\psi}(\mathbf{x}) \mathscr{F}(P_\theta, \mathrm{U})\, d\mathbf{x}. \quad (4.22)$$

Here, $p_D$ denotes the true data distribution, $y$ is the class, $\phi$, $\psi$ and $\theta$ are learnable parameters, while $\mathscr{F}$ corresponds to the chosen f-divergence. The gradient of the joint loss (4.22) w.r.t. the generator parameters $\psi$ vanishes in the first and the third term. The remaining terms enforce that the generated samples fool the discriminator and yield high-entropy closed-set predictions:

$$\frac{\partial L_{\text{adv}}(\phi,\psi;\theta)}{\partial \psi} = \frac{\partial}{\partial \psi} \int p_{G_\psi}(\mathbf{x}) \ln(1 - D_\phi(\mathbf{x}))\, d\mathbf{x} + \frac{\partial}{\partial \psi} \int p_{G_\psi}(\mathbf{x}) \mathscr{F}(P_\theta, \mathrm{U})\, d\mathbf{x}. \quad (4.23)$$

However, fooling the discriminator does not imply distributional coverage. In fact, the adversarial objective may cause mode collapse [113] which is detrimental to sample variability.

Our joint learning objective (4.20) optimizes the likelihood of inlier samples, the closed-set classification loss, and low confidence in synthetic negatives:

$$L(\zeta;\theta) = -\int p_D(\mathbf{x}) \ln p_\zeta(\mathbf{x})\, d\mathbf{x} - \int \sum_y p_D(y,\mathbf{x}) \ln P_\theta(y|\mathbf{x})\, d\mathbf{x} + \int p_\zeta(\mathbf{x}) \mathscr{F}(P_\theta, \mathrm{U})\, d\mathbf{x}. \quad (4.24)$$

Here, $p_\zeta$ is our normalizing flow. The gradient of the loss (4.24) w.r.t. the normalizing flow parameters $\zeta$ vanishes in the second term. The remaining terms enforce that the generated samples cover all modes of $p_D$ and, as before, yield high-entropy discriminative predictions:

$$\frac{\partial L(\zeta;\theta)}{\partial \zeta} = -\frac{\partial}{\partial \zeta} \int p_D(\mathbf{x}) \ln p_\zeta(\mathbf{x})\, d\mathbf{x} + \frac{\partial}{\partial \zeta} \int p_\zeta(\mathbf{x}) \mathscr{F}(P_\theta, \mathrm{U})\, d\mathbf{x}. \quad (4.25)$$

The resulting gradient entices the generative model to produce samples along the border of the inlier distribution. Hence, we say that our synthetic negatives are coverage-oriented. The presented analysis holds for any generative model that optimizes the density of the training data with the same set of parameters such as autoregressive models. Still, sampling with autoregressive models is slow, as discussed in previous chapters. Empirical confirmation of the described argument is in the experimental section of this thesis.

# Chapter 5

# Methodology

Experimental setups for generative modeling, dense anomaly detection and open-set segmentation require specialized datasets and benchmarks, that are described in Section 5.1. Quantitative performance evaluation on these datasets necessitates performance metrics elaborated in Section 5.2. The main implementation details relevant to the reproducibility of the reported results are described in Section 5.3.

## 5.1 Datasets and benchmarks

Our experimental evaluation is based on widely used datasets collected and annotated by third parties. Such datasets either include general crowdsourced images or collect application-specific ones. The former enables us to test the developed methods on general use cases with a variety of semantic objects, while the latter offers test protocols from specific real-world applications such as autonomous driving. We next describe particular setups used in the experimental evaluation of open-set segmentation.

### 5.1.1 Small image datasets

The **CIFAR-10** dataset [114] consists of 50k training images of resolution $32 \times 32$ pixels and 10k test images of the same resolution. The **ImageNet32** dataset [6] consists of over 1M training and 50k validation images divided across 1k classes. All images are resized to the resolution $32 \times 32$. Similarly, **ImageNet64** contains the same images resized to the resolution $64 \times 64$. The **CelebA** [115] dataset consists of 200k images associated with 10k identities. The dataset aggregates en face images of celebrities resized to the resolution $64 \times 64$. Figure 5.1 shows examples of low-resolution images from CIFAR-10. These datasets with small-resolution images are commonly used for benchmarking generative models.

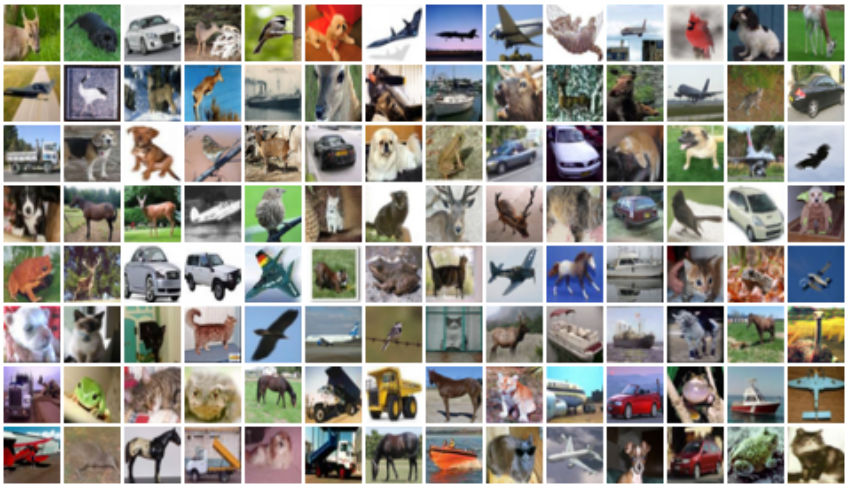

**Figure 5.1:** Example of small $32 \times 32$ images from the CIFAR10 dataset [114].

### 5.1.2 Crowdsourced datasets

The Common Objects in Context dataset [116] (COCO) collects and annotates 123k crowdsourced photos. The available annotations [117] contains 171 classes that are divided into 80 thing classes and 91 stuff classes. Thing classes often correspond to countable nouns (e.g. person or bike) while stuff classes correspond to uncountable nouns (e.g. sea or sky). The dataset contains 118k training and 5k validation images. We adapt this dataset for open-set segmentation by considering the 20 Pascal VOC [118] classes as known objects and the remaining 60 thing classes as unknowns. The stuff classes remain ignored. Our training set contains images without unknown classes and with at least one known class. Our test set contains all images with known and unknown thing classes. We refer to this setup as **COCO20/80**.

The Pascal VOC dataset [119] contains 13k images annotated with 20 semantic classes. We use this dataset for training and set all background training pixels to the mean pixel to prevent leakage of anomalous semantic content to the inlier representations. We use the 5k COCO validation images [116] with 133 classes for testing purposes. The unknown classes include 60 thing and 53 stuff classes. We refer to this setup as **Pascal-COCO**. Figure 5.2 visualizes three examples from COCO dataset.

**Figure 5.2:** Examples of images from the COCO dataset [116].

### 5.1.3 Traffic datasets

There are three main datasets that include traffic scenes: Cityscapes, Mapillary Vistas and WildDash. The **Cityscapes** dataset [4] contains 2.9k training images of traffic scenes collected in urban environments and annotated into 19 semantic classes. The **Mapillary Vistas** dataset [120] contains 19k training images collected across the world and annotations which can be related to the Cityscapes taxonomy. The **WildDash** dataset [65] contains 4.2k training images with adverse driving conditions. These datasets, visualized in Figure 5.3, are commonly used for training closed-set models. The trained models are then validated on well-established benchmarks we describe next.


**Figure 5.3:** Examples of images from Cityscapes [4], Mapillary Vistas [120], and WildDash2 [65]
.

The **Fishyscapes** benchmark [9] consists of two datasets: FS LostAndFound and FS Static. FS LostAndFound is a subset of original LostAndFound [121] that contains small objects on the roadway (e.g. toys, boxes or car parts that could fall off). FS Static contains Cityscapes validation images overlaid with Pascal VOC objects. The objects are positioned according to the camera perspective and further post-processed to obtain natural mixed-content images. Both datasets contain binary ground-truth labels with accurately segmented anomalies.

The **SegmentMeIfYouCan** benchmark (SMIYC) [66] consists of three datasets: AnomalyTrack, ObstacleTrack and LostAndFound-noKnown. AnomalyTrack and ObstacleTrack are created by curating real-world images and grouping them according to the anomaly sizes (large anomalies in AnomalyTrack and small anomalies on the road surface in ObstacleTrack). The LostAndFound-noKnown (LAF-noKnown) includes a selection of images from LostAndFound [121] where the anomalous objects do not correspond to the Cityscapes taxonomy. The SegmentMeIfYouCan benchmark supplies only binary ground-truth labels with accurately segmented anomalies. Additionally, we validate performance on the Cityscapes validation set by reinterpreting a subset of ignore classes as the unknown class, as proposed by [72]. More precisely, we consider all void Cityscapes classes except "unlabeled", "ego vehicle", "rectification border", "out of roi" and "license plate" as unknowns during validation.

The open-set performance can also be tested in simulated environments, as we showcase with the StreetHazards dataset. The **StreetHazards** dataset [15] is a synthetic dataset collected

with the CARLA game engine which simulates real-world environments. The simulated environments enable smooth anomaly injection and low-cost label extraction. Consequently, the dataset contains semantic per-pixel annotations of inlier content together with the unknowns.

## 5.2 Performance metrics

The success of a generative model in approximating the data distribution can be evaluated by measuring the average likelihood of the held-out test set. A high likelihood of the test dataset indicates a better approximation of the data distribution. Average likelihood of the test set is commonly reported in bits per dimension (BPD):

$$\text{BPD} = \frac{-\mathbb{E}_{\mathbf{x}} \log_2 p_\theta(\mathbf{x})}{C \cdot H \cdot W}. \tag{5.1}$$

Here, $W$ and $H$ correspond to image width and height, while $C$ is the number of channels. In the case of RGB images number of channels is 3.

The quality of samples produced by a generative model can be evaluated using Frechet Inception Distance (FID). Given a test dataset of size $N$, a generative model is sampled $N$ times to produce a set of artificial samples. All examples are then encoded in the feature space of the InceptionV3 [122] network pretrained on ImageNet. Finally, multivariate Gaussian is fitted on each of the two populations. The Frechet distance between the two Gaussians indicates the similarity between the two populations. Lower Frechet distance indicates a high quality of the generated samples.

Existing works [9, 66, 121] evaluate open-set segmentation through anomaly detection and closed-set segmentation metrics. The standard anomaly detection metrics are average precision, area under the ROC curve, and false-positive rate at true-positive rate of 95%. All three metrics assume anomalies are labeled as the positive class while regular examples are the negative class.

The average precision (AP) score summarizes the precision-recall curve as the weighted mean of precision achieved at every threshold:

$$\text{AP} = \sum_n (R_n - R_{n-1}) P_n, \tag{5.2}$$

where $R_n$ and $P_n$ are recall and precision at the $n$-th threshold. A higher value of AP indicates better separation between the two considered populations. Similarly, the area under the ROC curve (AUROC) summarizes the ROC curve into a scalar that describes the performance of a model for multiple thresholds. The false-positive rate at the true-positive rate of 95% ($\text{FPR}_{95}$) first selects the threshold which yields the true-positive rate of 95% and then measures the percentage of false positives among all negative samples. The lower $\text{FPR}_{95}$ score indicates better

performance. In our case, this metric can be interpreted as the probability of false anomaly detection for a threshold which correctly detects 95% of all anomalies.

Closed-set segmentation performance is measured by per-class intersection over union (IoU) score:

$$\mathrm{IoU}_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FP}_k + \mathrm{FN}_k}. \tag{5.3}$$

Here, $\mathrm{TP}_k$, $\mathrm{FP}_k$ and $\mathrm{FN}_k$ correspond to the number of true positives, false positives and false negatives for $k$-th class respectively. The per-class performances are then macro-averaged to obtain the scalar mIoU.

The existing performance metrics either ignore unknown classes, as in the case of closed-set mIoU, or collapse known classes into a single inlier class, as in the case of anomaly detection. However, none of these metrics clearly characterizes the impact of anomalies on segmentation performance in the open-set setup. Consequently, we propose a novel evaluation procedure for open-set segmentation which takes into account false positive semantic predictions at anomalies as well as false negative semantic predictions due to false anomaly detections. Our performance metric starts by thresholding the anomaly score so that it yields 95% TPR anomaly detection on held-out data. Then, we override the classification in pixels which score higher than the obtained threshold. This yields a recognition map with $K+1$ labels. We assess open-set segmentation performance according to a novel metric that we term open-mIoU. We compute open-IoU for the $k$-th class as follows:

$$\text{open-IoU}_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FP}_k^{\mathrm{os}} + \mathrm{FN}_k^{\mathrm{os}}}, \quad \text{where} \tag{5.4}$$

$$\mathrm{FP}_k^{\mathrm{os}} = \sum_{i=1, i\neq k}^{K+1} \mathrm{FP}_k^i, \quad \mathrm{FN}_k^{\mathrm{os}} = \sum_{i=1, i\neq k}^{K+1} \mathrm{FN}_k^i. \tag{5.5}$$

Different than the standard IoU formulation, open-IoU takes into account false predictions due to imperfect anomaly detection. In particular, a prediction of class $k$ at an outlier pixel (false negative anomaly detection) counts as a false positive for class $k$. Furthermore, a prediction of class K+1 at a pixel labelled as inlier class $k$ (false positive anomaly detection) counts as a false negative for class $k$. Note that we still average open-IoU over $K$ inlier classes. Thus, a recognition model with perfect anomaly detection gets assigned the same performance as in the closed world. This property would not be preserved if we averaged open-IoU over K+1 classes. Hence, a comparison between mIoU and open-mIoU quantifies the gap between the closed-set and open-set performance, unlike the related metrics [67, 72]. Also, open-IoU enables us to analyze false positive responses for specific classes, which is not possible by FPR at 95% TPR.

Figure 5.4 compares the considered closed-set (top left, $\mathrm{IoU}_k$) and open-set (right, open-$\mathrm{IoU}_k$) metrics. Imperfect anomaly detection impacts recognition performance through increased false positive and false negative semantics (designated in yellow and red, respectively). Differ-

ence between closed-set mIoU and open-mIoU reveals the performance gap due to inaccurate anomaly detection.

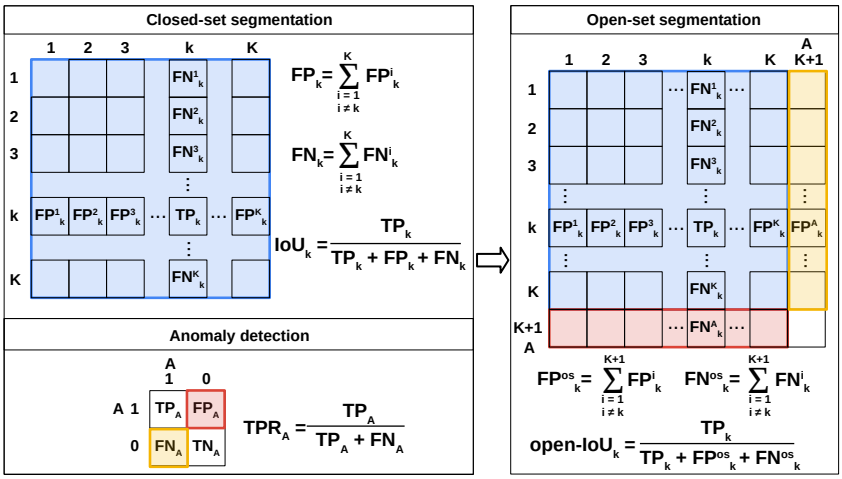


**Figure 5.4:** We extend the standard closed-set metric (top-left) with a novel open-set metric (right). Open-IoU takes into account false positive semantics at undetected anomalies as well as false negative semantics due to false positive anomalies. The proposed open-mIoU metric quantifies dense recognition performance in the presence of anomalies.

## 5.3 Implementation details

**DenseFlow.** We use the same DenseFlow-74-10 model in all experiments except ablations in order to illustrate the general applicability of our concepts. The first block of DenseFlow-74-10 uses 6 units with 5 glow-like modules in each DenseFlow unit, the second block uses 4 units with 6 modules, while the third block uses a single unit with 20 modules. We use the growth rate of 10 in all units. Each intra-module coupling starts with a projection to 48 channels. Subsequently, it includes a dense block with 7 densely connected layers, and the Nyström self-attention module with a single head. Since the natural images are discretized, we apply variational dequantization [100] to obtain continuous data which is suitable for normalizing flows. We train the proposed DenseFlow-74-10 architecture on ImageNet32 for 20 epoch using Adamax optimizer with learning rate set to $10^{-3}$ and batch size 64. We augment the training data by applying random horizontal flip with the probability of 0.5. We apply linear warm-up of the learning rate in the first 5000 iterations. During training, the learning rate is exponentially decayed by a factor of 0.95 after every epoch. The model is fine-tuned using a learning rate of $2 \cdot 10^{-5}$ for 2 epochs. Similarly, the model is trained for 10 epoch on ImageNet64, 50 epochs on CelebA and 580 epochs on CIFAR-10. In the case of CIFAR-10, we decay the learning rate by a factor of 0.9975. The model is fine-tuned for 1 epoch on ImageNet64, 5 epochs on CelebA and 70 epochs on CIFAR-10. We use batch size of 64 for CIFAR-10 and 32 for CelebA and ImageNet64. Other hyperparameters are the same as in ImageNet32 training.

**DenseHybrid.** We construct our open-set models by starting from any closed-set semantic segmentation model that trains with pixel-level cross-entropy loss. We implement the inlier posterior branch $g_\gamma$ as a trainable BN-ReLU-Conv1x1 module. We obtain unnormalized likelihood as the sum of exponentiated logits. We fine-tune the resulting open-set models on mixed-content images with pasted negative ADE20k instances or synthetic negative patches. In the case of SMIYC, we train LDN-121 [123] for 50 epochs in closed-set setup on images from Cityscapes [4], Vistas [120] and Wilddash2 [65] and fine tune for 10 epochs. In the case of Fishyscapes, we use DeepLabV3+ with WideResNet38 pretrained on Cityscapes [124]. We fine-tune the model for 10 epochs on Cityscapes. In the case of StreetHazards, we train LDN-121 for 120 epochs in the closed-world setting and then fine-tune the open-set model on mixed-content images. In the case of Pascal-COCO setup we train Segmenter with ViT-B/16 for 100 epochs on inlier data and then fine-tune the model for 10 epochs on real negatives. In the case of COCO20/80 we train Segmenter with ViT-B/16 for 80 epochs on inlier data and then fine-tune the model for 9 epochs on real negatives. In the case of synthetic negative data, we reduce the number of fine-tuning epochs to 5 to prevent overfitting. We optimize the loss (4.16) with the following hyperparameters: $\beta_1$ always equals 1. For traffic experiments with LDN-121 $\beta_2 = \beta_3 = 0.3$ and $\beta_4 = 0.03$. For DeepLabV3+ on traffic scenes $\beta_2 = \beta_3 = 0.1$ and $\beta_4 = 0.01$ except for Tbl. 6.4 where $\beta_2 = \beta_3 = 1.5$ and $\beta_4 = 0.15$. In the case of Pascal-COCO setup, $\beta_1 = 1$, $\beta_2 = \beta_3 = 1.5$, and $\beta_4 = 0.15$. Hyperparameter $\lambda$ from (4.20) always equals 0.03. Configurations that do not rely on real negative data leverage synthetic data of varying resolutions as generated by DenseFlow-45-6 [19]. All such experiments pre-train DenseFlow with the standard MLE loss on $64 \times 64$ crops from road-driving images (except for Pascal-COCO where we pre-train the flow on Pascal images) prior to joint learning. Our joint fine-tuning experiments last less than 24 hours on a single RTX A6000 GPU.

# Chapter 6

# Results

This chapter analyzes the performance of DenseFlow in generative modeling in Section 6.1. The performance of DenseHybrid in dense anomaly detection is in Section 6.2 while open-set segmentation results are in Section 6.3. In both cases, the reported results include models trained with and without real negative data. Contributions of stochastic skip connections are ablated in Section 6.4, while the benefits of hybrid anomaly score are validated in Section 6.5. Further experimental analysis considers computational requirements in Section 6.6 and practical aspects in Section 6.7. Qualitative results are visualized in Section 6.8.

## 6.1 Generative modeling

Table 6.1 compares the generative modeling performance of DenseFlow against contemporary normalizing flow architectures on four image datasets. DenseFlow attains more than 0.3 BPD better results on the two versions of the ImageNet dataset over the best baseline. In the case of the CIFAR10 dataset, DenseFlow equalizes the performance of VFlow in terms of BPD and attains second-best results in terms of FID. In the case of the CelebA dataset, DenseFlow attains over 1 BPD better results than the RealNVP baseline. These results indicate that equipping normalizing flow architecture with skip connections improves the model capacity and therefore generative modeling performance. Comparison of DenseFlow performance against alternative generative models such as autoregressive models, VAEs, GANs, diffusion models, and hybrid models are in the Appendix 8.6.

## 6.2 Pixel-level semantic anomaly detection

Table 6.2 presents dense anomaly detection performance of different methods on the Fishyscapes benchmark [13]. The top section considers models trained without real-world negative datasets while the bottom section collects the methods that require training on auxiliary negative datasets

**Table 6.1:** Generative modeling performance of DenseFlow on four image datasets.

| Method | CIFAR10 | | ImageNet32 | CelebA | ImageNet64 |
|---|---|---|---|---|---|
| | BPD | FID | BPD | BPD | BPD |
| Real NVP [98] | 3.49 | - | 4.28 | 3.02 | 3.98 |
| GLOW [99] | 3.35 | 46.90 | 4.09 | - | 3.81 |
| Wavelet Flow [125] | - | - | 4.08 | - | 3.78 |
| Residual Flow [102] | 3.28 | 46.37 | 4.01 | - | 3.78 |
| i-DenseNet [126] | 3.25 | - | 3.98 | - | - |
| Flow++ [100] | 3.08 | - | 3.86 | - | 3.69 |
| ANF [108] | 3.05 | **30.60** | 3.92 | - | 3.66 |
| VFlow [107] | **2.98** | - | 3.83 | - | 3.66 |
| DenseFlow (ours) | **2.98** | 34.90 | **3.63** | **1.99** | **3.35** |

such as ImageNet or ADE20k. We also denote methods which rely on image resynthesis since such approaches require considerable computational requirements that preclude efficient inference. Following [11, 66], we use DeepLabV3+ [124] segmentation model trained on Cityscapes. DenseHybrid trained with synthetic negative data (SynDenseHybrid) achieves 20% and 10% absolute performance improvements in terms of AP and $FPR_{95}$ over the best previous result on the FS LostAndFound dataset. In the case of the FS Static dataset, SynDenseHybird achieves 13% improvement in terms of $FPR_{95}$ over the best baseline while attaining the second-best average precision score.

Among methods that train on real negative datasets, DenseHybrid outperforms the best previous result on the FS LostAndFound dataset by 9% in terms of $FPR_{95}$ while achieving marginal improvements in terms of average precision score. In the case of FS Static, DenseHybrid improves the best previous result by 14% in terms of $FPR_{95}$ while achieving the second-best average precision score. Furthermore, DenseHybrid consistently outperforms all baselines that do not rely on image-resynthesis. We also note that all methods based on DeepLabV3+ attain comparable closed-set classification performance, as shown in the rightmost table column.

Next, we analyze dense anomaly detection performance on the SegmentMeIfYouCan benchmark [66]. Table 6.3 presents anomaly detection performance on the tree datasets of SegmentMeIfYouCan. SynDenseHybrid consistently outperforms all baselines that avoid training on real negative data and image resynthesis. Most notably, SynDenseHybrid improves the best results by 12% and 5% in terms of $FPR_{95}$ score on ObstacleTrack and LAF-noKnown respectively. Compared to resynthesis-based approaches, SynDenseHybrid yields lower false positive rates on ObstacleTrack and LAF-noKnown while attaining only slightly lower average precision scores. Note that SynDenseHybrid attains these results while avoiding image resynthesis during the inference, enabling real-time inference.

**Table 6.2:** Dense anomaly detection on the Fishyscapes benchmark [13].

| Method | Auxiliary data | Image resynthesis | FS LAF AP | FS LAF $FPR_{95}$ | FS Static AP | FS Static $FPR_{95}$ | Cityscapes val mIoU |
|---|---|---|---|---|---|---|---|
| Image Resynthesis [50] | ✗ | ✓ | 5.7 | 48.1 | 29.6 | 27.1 | 81.4 |
| Max softmax [24] | ✗ | ✗ | 1.8 | 44.9 | 12.9 | 39.8 | 80.3 |
| SML [46] | ✗ | ✗ | 31.7 | 21.9 | 52.1 | 20.5 | - |
| Embedding Density [9] | ✗ | ✗ | 4.3 | 47.2 | **62.1** | 17.4 | 80.3 |
| SynDenseHybrid (ours) | ✗ | ✗ | **51.8** | **11.5** | 54.7 | **15.5** | 79.9 |
| SynBoost [52] | ✓ | ✓ | 43.2 | 15.8 | **72.6** | 18.8 | 81.4 |
| Prior Entropy [47] | ✓ | ✗ | 34.3 | 47.4 | 31.3 | 84.6 | 70.5 |
| Void Classifier [9] | ✓ | ✗ | 10.3 | 22.1 | 45.0 | 19.4 | 70.4 |
| Dirichlet prior [47] | ✓ | ✗ | 34.3 | 47.4 | 84.6 | 30.0 | 70.5 |
| DenseHybrid (ours) | ✓ | ✗ | **43.9** | **6.2** | **72.3** | **5.5** | 81.0 |

Amongst the methods that require real negative data during the training, DenseHybrid outperforms all baselines by a wide margin. For example, DenseHybrid attains 20% and 16% improvement over the best baseline in terms of AP on AnomalyTrack and ObstacleTrack datasets. A single exception is SynBoost which attains marginally better results in terms of AP on the LAF-noKnown dataset by relying on computationally heavy image resynthesis. Still, DenseHybrid consistently outperforms all baselines in terms of false-positive rate.

**Table 6.3:** Dense anomaly detection on the SegmentMeIfYouCan benchmark [66].

| Method | Auxiliary data | Image resynthesis | AnomalyTrack AP | AnomalyTrack $FPR_{95}$ | ObstacleTrack AP | ObstacleTrack $FPR_{95}$ | LAF-noKnown AP | LAF-noKnown $FPR_{95}$ |
|---|---|---|---|---|---|---|---|---|
| Image Resynthesis [50] | ✗ | ✓ | **52.3** | **25.9** | 37.7 | 4.7 | 57.1 | 8.8 |
| Road Inpaint. [127] | ✗ | ✓ | - | - | 54.1 | 47.1 | **82.9** | 35.8 |
| JSRNet [53] | ✗ | ✓ | 33.6 | 43.9 | 28.1 | 28.9 | 74.2 | 6.6 |
| Max softmax [24] | ✗ | ✗ | 28.0 | 72.1 | 15.7 | 16.6 | 30.1 | 33.2 |
| MC Dropout [26] | ✗ | ✗ | 28.9 | 69.5 | 4.9 | 50.3 | 36.8 | 35.6 |
| ODIN [25] | ✗ | ✗ | 33.1 | 71.7 | 22.1 | 15.3 | 52.9 | 30.0 |
| Mahalanobis [128] | ✗ | ✗ | 20.0 | 87.0 | 20.9 | 13.1 | 55.0 | 12.9 |
| Embedding Density [9] | ✗ | ✗ | 37.5 | 70.8 | 0.8 | 46.4 | 61.7 | 10.4 |
| SynDenseHybrid (ours) | ✗ | ✗ | **51.5** | **33.2** | **64.0** | **0.6** | **78.8** | **1.1** |
| SynBoost [52] | ✓ | ✓ | 56.4 | 61.9 | 71.3 | 3.2 | **81.7** | 4.6 |
| Void Classifier [9] | ✓ | ✗ | 36.6 | 63.5 | 10.4 | 41.5 | 4.8 | 47.0 |
| DenseHybrid (ours) | ✓ | ✗ | **78.0** | **9.8** | **87.1** | **0.2** | **78.7** | **2.1** |

We next validate our method by considering a subset of Cityscapes void classes as the un-

known class. Table 6.4 compares performance according to the AUROC metric denoted AUC column. SynDenseHybrid attains absolute improvement of four percentage points over the previous best approach OpenGAN [72]. We do not report results when training on real negative data since previous works refrain from such training setup.

**Table 6.4:** Anomaly detection on Cityscapes val with a subset of ignore classes considered as unknowns.

| Method | AUC | Method | AUC | Method | AUC |
|---|---|---|---|---|---|
| MSP [24] | 72.1 | GDM [128] | 74.3 | Entropy [129] | 69.7 |
| GMM | 76.5 | OpenMax [58] | 75.1 | K+1 classifier | 75.5 |
| C2AE [130] | 72.7 | OpenGAN-O [72] | 70.9 | ODIN [25] | 75.5 |
| OpenGAN [72] | 88.5 | MC dropout [26] | 76.7 | SynDenseHybrid (ours) | **92.9** |

## 6.3 Open-set segmentation

We recover open-set segmentation by fusing a closed-set segmentation with thresholded dense anomaly detection, as described in Chapter 4. We measure open-set performance according to mean $F_1$ ($\overline{F_1}$) score and the proposed open-mIoU ($\overline{\text{oIoU}}$) metric. Table 6.5 presents performance evaluation on the StreetHazards dataset. The left part of the table considers semantic anomaly detection while the right part considers closed-set and open-set segmentation. Our method outperforms contemporary approaches in anomaly detection. For example, SynDenseHybrid outperforms the best baseline DML [59] by 5% in terms of average precisions while DenseHybrid outperforms the best baseline Outlier head [12] by 10%. Furthermore, our method achieves the best open-set performance (columns $\overline{\text{oIoU}}$ and $\overline{F_1}$) despite a moderate capacity of LDN-121 ($\overline{\text{IoU}}$ column). The last column quantifies the performance gap between closed-set and open-set performance as the difference between IoU and oIoU. Our method achieves the least performance gap of around 18%. Nevertheless, an ideal model would deliver equal open-set and closed-set metrics. Hence, we conclude that the current state of the art is incapable of delivering closed-set performance in open-set setups. Note that Table 6.5 does not list ObsNet [76] since they aim to detect classification errors instead of anomalies.

Table 6.6 presents open-set segmentation performance on crowdsourced photos from the COCO dataset. In the case of the Pascal-COCO setup, SynDenseHybrid consistently outperforms previous approaches [15, 25, 62] both in anomaly detection and open-set segmentation. Most notably, SynDenseHybrid attains 20% absolute improvement over the best baseline in terms of open-IoU. In the case of anomaly detection on the COCO20/80 setup, SynDenseHybrid outperforms baselines in terms of AUROC and false-positive rate while achieving the second-best average precision. In the case of open-set segmentation, SynDenseHybrid consistently

**Table 6.5:** Performance evaluation on StreetHazards [15]. We evaluate anomaly detection (Anomaly), closed-set (Clo.) and open-set segmentation (Open-set), as well as the open-set performance gap (Gap).

| Method | Anomaly | | | Closed-set | Open-set | | Gap |
|---|---|---|---|---|---|---|---|
| | AP | $FPR_{95}$ | AUROC | $\overline{\text{IoU}}$ | $\overline{F_1}$ | $o\overline{IoU}$ | |
| SynthCP [51] | 9.3 | 28.4 | 88.5 | - | - | - | - |
| Dropout [26] | 7.5 | 79.4 | 69.9 | - | - | - | - |
| TRADI [131] | 7.2 | 25.3 | 89.2 | - | - | - | - |
| SO+H [73] | 12.7 | 25.2 | 91.7 | 59.7 | - | - | - |
| DML [59] | 14.7 | 17.3 | 93.7 | - | - | - | - |
| MSP [24] | 7.5 | 27.9 | 90.1 | 65.0 | 46.4 | 35.1 | 29.9 |
| ODIN [25] | 7.0 | 28.7 | 90.0 | 65.0 | 41.6 | 28.8 | 36.2 |
| ReAct [37] | 10.9 | 21.2 | 92.3 | 62.7 | 46.4 | 34.0 | 28.7 |
| SynDenseHybrid (ours) | **19.7** | **17.4** | **93.9** | 61.3 | **50.6** | **37.3** | **24.0** |
| Energy [32] | 12.9 | 18.2 | 93.0 | 63.3 | 50.4 | 42.7 | 29.9 |
| OE [31] | 14.6 | 17.7 | 94.0 | 61.7 | 56.1 | 43.8 | 17.9 |
| OH [12] | 19.7 | 56.2 | 88.8 | **66.6** | - | 33.9 | 32.7 |
| OH*MSP [10] | 18.8 | 30.9 | 89.7 | **66.6** | - | 43.6 | 23.0 |
| DenseHybrid (ours) | **30.2** | **13.0** | **95.6** | 63.0 | **59.7** | **45.8** | **17.2** |

outperforms baselines on the two considered metrics. Most notably, performance improvement is 8% in terms of macro $F_1$ score over the best baseline ODIN [25].

We compare DenseHybrid performance trained with real negative samples from ADE20k with previous approaches [31, 32] trained in the same setup. In the case of the Pascal-COCO setup, DenseHybrid consistently outperforms previous approaches in anomaly detection and open-set segmentation. For instance, absolute performance improvement over the best baseline is 7% in terms of open-IoU and 3% in terms of false-positive rate. In the case of the COCO20/80 setup, DenseHybid outperforms alternative approaches in open-set segmentation and two out of three metrics in anomaly detection. Most notably, DenseHybrid attains 5% improvement in terms of open-IoU and 4% in terms of false-positive rate.

Closed-set models reach more than 90% mIoU in the case of Pascal-COCO and over 75% in the case of COCO20/80, but open-IoU peaks at 41% and 16%. Analysis of false positives reveals that the task is hard due to large intra-class variation (e.g. different species of potted plants). Moreover, some unknown classes have a similar appearance to known classes (eg. unknown *zebra* and known *horse*). Finally, the benchmark has high openness [56]: there are more than 3× unknown than known classes. Interestingly, synthetic negatives prevail in the case of COCO20/80. This indicates that ADE20k negatives may not be an adequate negative dataset for this setup. Finally, we noticed that many mistakes of DenseHybrid coincide with

labelling errors, as we will show in the qualitative results.

**Table 6.6:** Open-set segmentation on crowdsourced photos from COCO val.

| Method | Pascal - COCO | | | | | COCO 20/80 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AP | AUC | $FPR_{95}$ | $\overline{F_1}$ | $\overline{oIoU}$ | AP | AUC | $FPR_{95}$ | $\overline{F_1}$ | $\overline{oIoU}$ |
| ML [15, 62] | 93.7 | 82.2 | 56.5 | 22.0 | 13.2 | **63.7** | 75.4 | 63.5 | 15.6 | 9.8 |
| ODIN [25] | 90.3 | 75.0 | 66.1 | 13.3 | 6.5 | 54.6 | 70.6 | 61.4 | 15.8 | 10.5 |
| SynDenseHybrid | **95.1** | **86.0** | **46.8** | **48.4** | **33.9** | 62.4 | **77.1** | **59.7** | **24.2** | **16.3** |
| Energy [32] | 94.2 | 83.5 | 54.6 | 19.3 | 11.4 | **63.7** | 74.7 | 65.7 | 14.7 | 8.9 |
| OE [31] | 95.9 | 88.0 | 46.0 | 49.1 | 34.1 | 64.7 | 75.4 | 68.7 | 13.5 | 7.7 |
| DenseHybrid | **96.5** | **89.2** | **43.0** | **55.3** | **41.0** | 59.4 | **75.6** | **61.7** | **22.8** | **14.7** |

## 6.4 Ablating components of DenseFlow

Table 6.7 explores the contributions of incremental augmentation of latent variables with random noise, stochastic skip connections, and dense connectivity in the internal networks of coupling layers. The latter compares intra-module coupling network based on the fusion of fast self-attention and a densely connected convolutional block with the original Glow coupling [99].

The bottom row of the table corresponds to a DenseFlow-45-6 model. The first DenseFlow block has 5 DenseFlow units with 3 invertible modules per unit. The second DenseFlow block has 3 units with 5 modules, while the final block has 15 modules in a single unit. We use the growth rate of 6. The top row of the table corresponds to the standard normalized flow [98, 99] with three blocks and 15 modules per block. Consequently, all models have the same number of invertible glow-like modules. All models are trained on CIFAR10 for 300 epochs and then fine-tuned for 10 epochs. We use the same training hyperparameters for all models. The proposed cross-unit coupling improves the density estimation from 3.42 bpd to 3.37 bpd (row 3) starting from a model with the standard glow modules (row 1). When a model is equipped with our intra-module coupling, cross-unit coupling leads to improvement from 3.14 bpd (row 4) to 3.07 bpd (row 6). Hence, the proposed cross-unit coupling improves the density estimation in all experiments. Similarly, models with stochastic skip connections outperform models with simple random noise (row 2 vs row 3, and row 5 vs row 6).

**Table 6.7:** Validation of DenseFlow components on the CIFAR10 dataset.

| # | Latent variable augmentation | Stochastic skip connections | Dense connectivity within coupling layers | BPD |
|---|---|---|---|---|
| 1 | ✗ | ✗ | ✗ | 3.42 |
| 2 | ✓ | ✗ | ✗ | 3.40 |
| 3 | ✓ | ✓ | ✗ | 3.37 |
| 4 | ✗ | ✗ | ✓ | 3.14 |
| 5 | ✓ | ✗ | ✓ | 3.08 |
| 6 | ✓ | ✓ | ✓ | 3.07 |

## 6.5 Ablating components of DenseHybrid

Table 6.8 validates components of our hybrid approach on Fishyscapes val. The top two sections validate the two DenseHybrid components, $\hat{p}(\mathbf{x})$ and $P(d_{\text{in}}|\mathbf{x})$, when training on real and synthetic negative data, respectively. We observe that the hybrid score outperforms unnormalized density which outperforms inlier posterior. We observe the same quantitative behaviour when training on real and synthetic negative data. The bottom section replaces our unnormalized likelihood with pre-logit likelihood estimates by a normalizing flow. The flow is applied point-wise in order to obtain dense likelihood [9]. This can also be viewed as a generalization of a previous image-wide open-set approach [45] to dense prediction. We still train on negative data in an end-to-end fashion in order to make the two generative components comparable. The resulting model delivers good performance on FS Static and poor performance on FS LostAndFound. We attribute better performance of our unnormalized density (4.7) with respect to the point-wise flow due to $4\times$ subsampling of the pre-logits to which the flow was fitted. Moreover, our unnormalized density ensures much faster inference due to lower computational complexity.

Table 6.9 shows open-set segmentation performance depending on the choice of the anomaly detector on crowdsourced photos. Generative and discriminative components of our approach yield comparable open-set performance, while their ensemble achieves substantial further improvement. A detailed analysis shows that generative and discriminative detectors are only moderately correlated. In the case of Pascal-COCO we have $\rho = 0.59$, $\alpha = 1.22$, $e = 1.09$, $C_1 = 0.42$, $C_2 = 0.18$, while in the case of COCO20/80 we have $\rho = 0.56$, $\alpha = 1.44$, $e = 1.22$, $C_1 = 0.7$, $C_2 = 0.04$. Hence, the condition (4.4) is satisfied. Note that it makes no sense to ensemble two arbitrary anomaly detectors since they are often well-correlated (e.g. max-logit [15] and free-energy [32] have $\rho$=0.98), which again supports our approach.

Next, we ablate the loss used for training generative model for synthetic negatives and note that arbitrary loss may not be sufficient. In particular, requiring synthetic negatives to stand out from the inliers may be easier to overfit than requiring them to produce a uniform prediction

**Table 6.8:** Validation of dense hybrid anomaly detection on Fishyscapes val. Our method outperforms its generative and discriminative components.

| Anomaly detector | Neg. data | FS LAF | | FS Static | |
|---|---|---|---|---|---|
| | | AP | $FPR_{95}$ | AP | $FPR_{95}$ |
| Discriminative $(1-P(d_{\text{in}}\|\mathbf{x}))$ | Real | 46.5 | 38.3 | 53.5 | 30.9 |
| Generative $\hat{p}(\mathbf{x}) = \text{LSE}(\mathbf{s})$ | Real | 58.2 | 7.3 | 58.0 | 5.3 |
| Hybrid $(1-P(d_{\text{in}}\|\mathbf{x}))/\hat{p}(\mathbf{x})$ | Real | **60.5** | **6.0** | **63.1** | **4.2** |
| Discriminative $(1-P(d_{\text{in}}\|\mathbf{x}))$ | Synthetic | 30.1 | 35.0 | 48.8 | 39.8 |
| Generative $\hat{p}(\mathbf{x}) = \text{LSE}(\mathbf{s})$ | Synthetic | 58.1 | 9.0 | 44.6 | 9.5 |
| Hybrid $(1-P(d_{\text{in}}\|\mathbf{x}))/\hat{p}(\mathbf{x})$ | Synthetic | **60.2** | **7.9** | **52.1** | **7.7** |
| Gen. flow $p(\mathbf{z})$ | Real | 5.7 | 58.9 | 61.7 | 7.6 |
| Hybrid $(1-P(d_{\text{in}}\|\mathbf{x}))/p(\mathbf{z})$ | Real | 6.5 | 46.1 | 65.1 | 6.5 |

**Table 6.9:** Validation of DenseHybrid components on COCO val.

| Anomaly detector | Pascal-COCO | | COCO20/80 | |
|---|---|---|---|---|
| | $\overline{\text{oIoU}}$ | $\overline{F_1}$ | $\overline{\text{oIoU}}$ | $\overline{F_1}$ |
| Generative $\hat{p}(\mathbf{x}) = \text{LSE}(\mathbf{s})$ | 38.1 | 51.3 | 16.7 | 22.9 |
| Discriminative $(1-P(d_{\text{in}}\|\mathbf{x}))$ | 38.6 | 52.8 | 14.4 | 21.8 |
| Hybrid $(1-P(d_{\text{in}}\|\mathbf{x}))/\hat{p}(\mathbf{x})$ | **42.0** | **55.3** | **17.8** | **24.2** |

over 19 classes. According to our intuition, the latter should provide a better learning signal than the former. Table 6.10 experimentally validates our intuition and shows clear performance gains of requiring uniform classification.

**Table 6.10:** Validation of the learning objective for normalizing flow that generates synthetic negatives.

| Loss type | FS LAF | | FS Static | |
|---|---|---|---|---|
| | AP | $FPR_{95}$ | AP | $FPR_{95}$ |
| $L_{\text{mle}} + L_{\text{d}} + L_{\text{x}}^{\text{UB}}$ | 46.1 | 12.4 | 41.8 | 12.1 |
| $L_{\text{mle}} + L_{\text{JSD}}$ | 60.2 | 7.9 | 52.1 | 7.7 |

Table 6.11 validates different sources of negative data. We compare the synthetic negatives from Figure 6.8 with patches of uniform noise, local adversarial attacks [76], inlier crops [53] as well as samples from a jointly trained GAN [71]. We include the average AP over four datasets (last column) as a metric of overall anomaly detection performance. The top section of the table indicates that our synthetic negatives outperform all alternative approaches and come close to real negative data. In particular, we observe significant performance improvement over GAN negatives, as hypothesised in Section 4.4.1. The bottom section of the table compares real

negatives from ADE20k [10] with 10.8k samples from a pre-trained conditional diffusion model [132]. We have used prompts of the form "A photograph of *cls*" where *cls* stands for a random class description from ADE20k. Note that diffusion negatives cannot be directly compared with other synthetic approaches due to training on the huge LAION2B dataset. Furthermore, the design of textual prompts for generating synthetic negative data is still an open problem, which requires further work that is out of the scope of our work.

**Table 6.11:** DenseHybrid performance with different kinds of negative data.

| Source of | FS-val | | SMIYC-val | | Average |
|---|---|---|---|---|---|
| negatives | LAF | Stat. | Anom. | Obs. | |
| Uniform noise | 56.9 | 37.4 | 70.5 | 3.5 | 42.1 |
| Inlier crops [53] | 64.3 | 36.2 | 77.2 | 62.8 | 60.1 |
| Loc. Adv. Attacks [76] | 44.5 | 36.8 | 78.9 | 62.2 | 55.6 |
| GAN [71] | 60.9 | 38.8 | 72.8 | 44.9 | 54.4 |
| Jointly trained NF | 60.2 | 46.0 | 77.7 | 86.2 | **67.5** |
| ADE20k-instances [10] | 63.7 | 68.4 | 76.2 | 86.0 | **73.6** |
| ADE20k-crops | 62.1 | 47.7 | 76.6 | 74.2 | 65.2 |
| ADE20k-mix | 63.5 | 61.9 | 76.6 | 87.0 | 72.3 |
| Stable-diffusion [132] | 70.4 | 57.9 | 76.8 | 49.3 | 63.6 |

Figure 6.1 shows anomaly detection performance when mixing real negatives from ADE20k and synthetic negatives generated by our normalizing flow. The negative data is mixed according to the hyperparameter $b$ as described in Sec. 4.4. We observe varying performance for different values of $b$. Still, the best performance is achieved when we train solely on real negatives ($b$=1). Investigating more advanced procedures for mixing real and synthetic negative data is an interesting direction for future work.

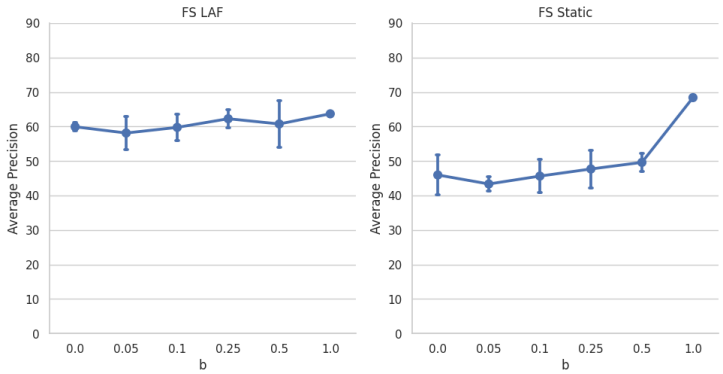


**Figure 6.1:** Performance of our hybrid anomaly detector when training on mixtures of real and synthetic negatives for $b \in [0, 1]$.

## 6.6 Computational overhead

Real-time inference is an absolute necessity for many real-world applications. Thus, we analyze the computational overhead of DenseHybrid over the standard closed-set model. Table 6.12 compares computational overheads of DenseHybrid with prominent anomaly detectors on two-megapixel images. All measurements are averaged over 200 runs on RTX3090. DenseHybrid involves a negligible computational overhead of 0.1 GFLOPs and 2.8 miliseconds. These experiments indicate that DenseHybrid establishes as a new strong baseline for outlier-aware real-time inference. In addition, we observe that the image resynthesis is not applicable for real-time inference on present hardware.

**Table 6.12:** Computational overhead of prominent anomaly detectors over the baseline semantic segmentation model when inferring on two-megapixel images. DenseHybrid introduces only minimal computational overhead over the closed-set model. The inference time is in milliseconds.

| Method | Resynthesis | Inference time (ms) | FPS | GFLOPs |
|---|---|---|---|---|
| SynBoost [52] | ✓ | 1055.5 | <1 | - |
| SynthCP [51] | ✓ | 146.9 | <1 | 4551.1 |
| LDN-121 [123] | ✗ | 60.9 | 16.4 | 202.3 |
| LDN-121 + SML [46] | ✗ | 75.4 | 13.3 | 202.6 |
| LDN-121 + DH (ours) | ✗ | **63.7** | **15.7** | **202.4** |

## 6.7 Anomaly detection depending on the distance

Road driving scenes typically involve a wide range of depth. Hence, we explore the anomaly detection performance at different ranges from the camera in order to gain a better performance insight. We perform these experiments on the LostAndFound test set [121] since it provides information about the depth in each ground pixel. Due to errors in the provided disparity maps, we perform our analysis up to 50 meters from the camera. Table 6.13 compares DenseHybrid against previous works [15, 24, 52]. DenseHybrid outperforms all baselines and achieves strong results even at large distances from the camera. A single exception is SynBoost [52] which attains slightly better performance than DenseHybrid at the shortest range. However, the computational complexity of SynBoost caused by image resynthesis precludes real-time deployment on present hardware.

## 6.8 Qualitative results

Figure 6.2 shows generated images using DenseFlow-74-10 trained on CelebA.

**Table 6.13:** Anomaly detection performance at different distances from the camera.

| Range (m) | MSP [24] | | ML [15] | | SynBoost [52] | | DH (ours) | |
|---|---|---|---|---|---|---|---|---|
| | AP | $FPR_{95}$ | AP | $FPR_{95}$ | AP | $FPR_{95}$ | AP | $FPR_{95}$ |
| 5-10 | 28.7 | 16.4 | 76.1 | 5.4 | **93.7** | **0.2** | 90.7 | 0.3 |
| 10-15 | 28.8 | 29.7 | 73.9 | 16.2 | 78.7 | 17.7 | **89.8** | **1.1** |
| 15-20 | 26.0 | 28.8 | 78.2 | 5.9 | 76.9 | 25.0 | **92.9** | **0.6** |
| 20-25 | 25.1 | 44.2 | 69.6 | 12.8 | 70.0 | 23.3 | **89.1** | **1.4** |
| 25-30 | 29.0 | 41.3 | 72.6 | 9.5 | 65.6 | 18.8 | **89.5** | **1.4** |
| 30-35 | 26.2 | 47.8 | 70.2 | 10.0 | 58.5 | 27.4 | **87.7** | **2.5** |
| 35-40 | 29.6 | 44.7 | 71.0 | 9.8 | 59.8 | 25.4 | **85.0** | **3.7** |
| 40-45 | 31.7 | 43.2 | 74.0 | 9.8 | 60.0 | 25.8 | **85.6** | **4.7** |
| 45-50 | 33.7 | 45.3 | 73.9 | 11.0 | 53.3 | 29.9 | **82.1** | **6.3** |

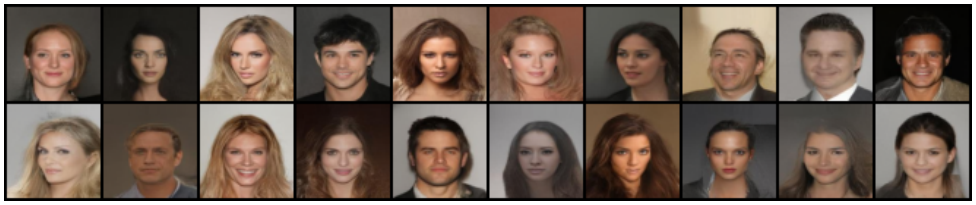

**Figure 6.2:** Samples from DenseFlow-74-10 trained on CelebA.

Figure 6.3 shows the semantic anomaly detection performance of DenseHybrid on five traffic datasets that we consider. The top row shows the input RGB images. The bottom row shows anomaly maps produced by DenseHybrid. The detected anomalous pixels are designated in yellow. Accurate dense anomaly detection performance enables us to deliver competitive open-set segmentation.

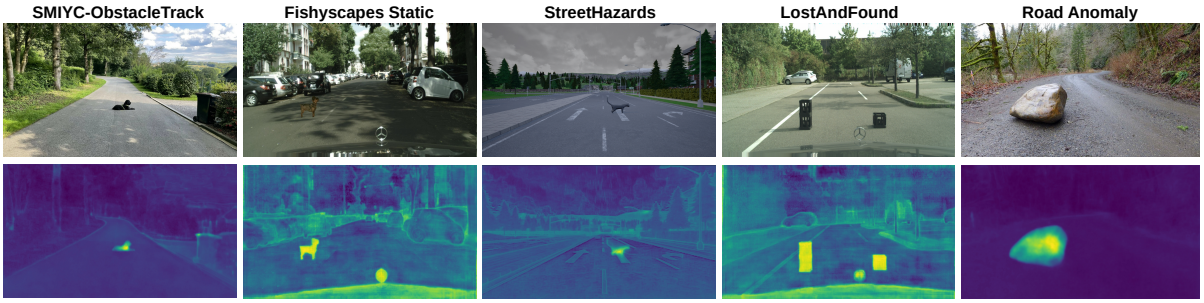


**Figure 6.3:** Qualitative performance of the proposed DenseHybrid approach on standard datasets. Top: input images. Bottom: dense maps of the proposed anomaly score. Unknown pixels are assigned with higher anomaly scores as designated in yellow.

Figure 6.4 visualizes qualitative open-set segmentation performance on the StreetHazards test set. The first row shows input RGB images. The second row shows anomaly maps produced by DenseHybrid. The third row includes open-set segmentation output produced by combining dense anomaly detection (second row) with closed-set segmentation. For reference, the fourth

row shows open-set segmentation with an alternative energy-based approach [32] that yields more false positives at TPR=95%, as designated with red rectangles. The last row includes ground truth opne-set segmentation maps.

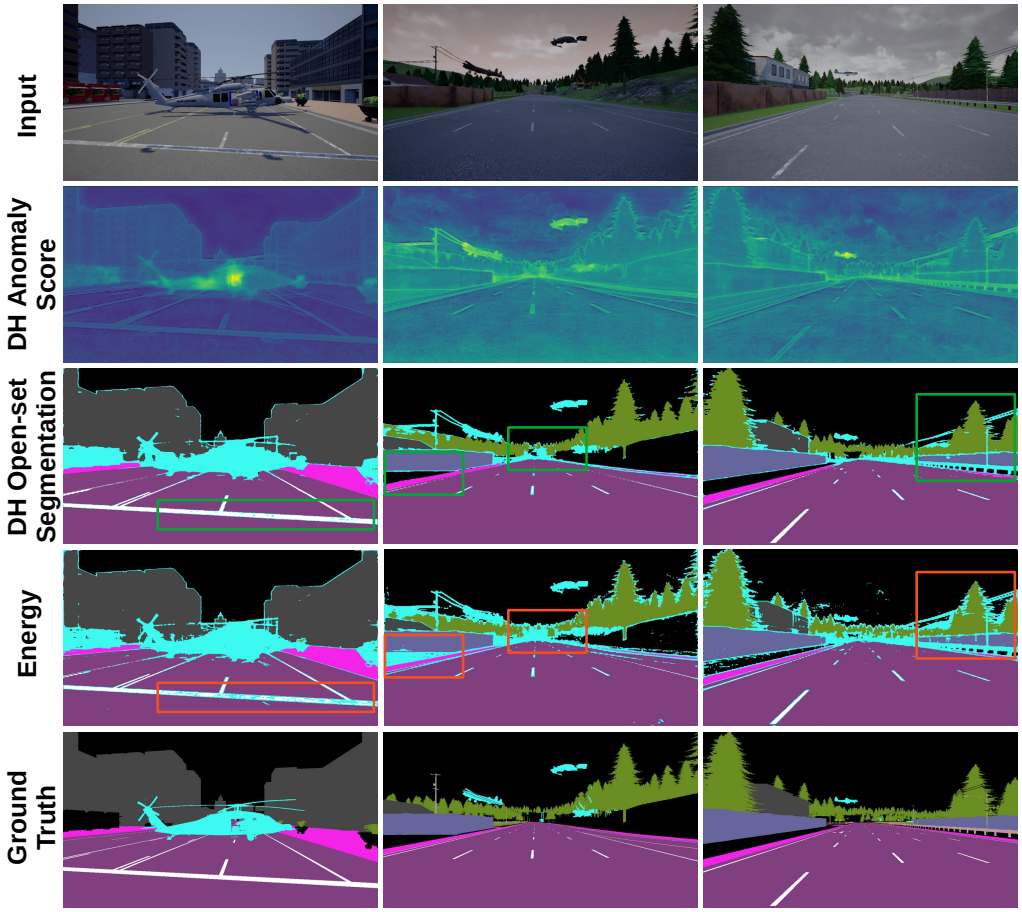


**Figure 6.4:** Qualitative open-set segmentation performance on StreetHazards. DenseHybrid delivers a more accurate open-set performance with respect to the energy-based approach [32], as denoted with red rectangles.

Figure 6.5 shows qualitative open-set segmentation performance, based on different anomaly detectors. Since we set the anomaly threshold to 95% of the true-positive rate, performance improvements are indicated by a lower count of false positives. The leftmost column shows input RGB images. The next two columns show open-set segmentation performance for discriminative and generative component of DenseHybrid. The fourth column shows open-set segmentation built on our hybrid anomaly detector. Poorly segmented regions are denoted with red rectangles while green rectangles denote more accurate segmentation. We observe the fusing potential of the hybrid anomaly detector. Ground truth labels are in the rightmost column.

Figure 6.6 shows open-set segmentation with DenseHybrid on the COCO20/80 dataset. The left column shows the input RGB image. The center column shows open-set segmentation with DenseHybrid. For visualization purposes, we override predictions in ignore pixels and colour them dark. The right column shows ground-truth labels. We observe that DenseHybrid can

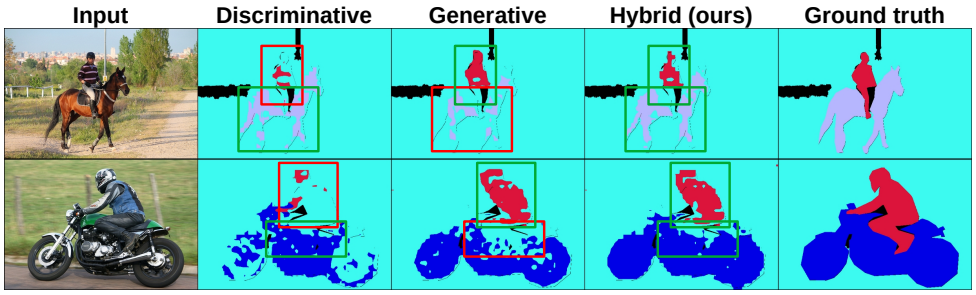


**Figure 6.5:** Open-set segmentation on Pascal-COCO with discriminative, generative and hybrid anomaly detector. Red and green boxes indicate the abundance and absence of false positive anomalies.

detect both large and small unknowns.

Figure 6.7 shows open-set segmentation performance depending on the choice of anomaly detector on Fishyscapes LostAndFound. The leftmost column shows the RGB input image. The second column shows open-set segmentation with our dense hybrid anomaly detector. Compared with its generative and discriminative components visualized in the two rightmost columns, the hybrid anomaly detector yields the lowest false-positive count for TPR $= 95\%$.

Figure 6.8 shows synthetic negatives produced by joint training of dense classifier and normalizing flow, as described in Section 4.4. The generated samples vary in spatial resolution and lack meaningful visual concepts. Interestingly, training our open-set model on such samples yields only slightly worse performance than the model trained on real negative data from the ADE20k dataset.

Figure 6.9 shows synthetic negatives produced by local adversarial attacks [76], jointly trained GAN [71], and the proposed normalizing flow. We observe that adversarial attacks appear as blurred patches of inlier scenes. This may make them an inadequate proxy for test-time anomalies due to requiring a lot of capacity to discriminate them from the inliers [133]. Similarly, it is well known that GAN samples struggle to achieve visual variety [48]. Contrary, our normalizing flow produces negatives which differ from the inlier scenes and are visually diverse. Different from our jointly trained normalizing flow, the text-to-image models cannot produce dataset-specific samples and require textual descriptions. Consequently, they are not in the same ballpark as our method. Furthermore, the design of textual prompts for generating synthetic negative data is still an open problem, which requires further work that is out of the scope of our work. Note that the quantitative comparison of the three approaches can be found in Table 6.11.


**Figure 6.6:** Qualitative examples of open-set segmentation on COCO20/80. Semantic anomalies are denoted in cyan. For visualization purposes, we override predictions in void pixels (dark).

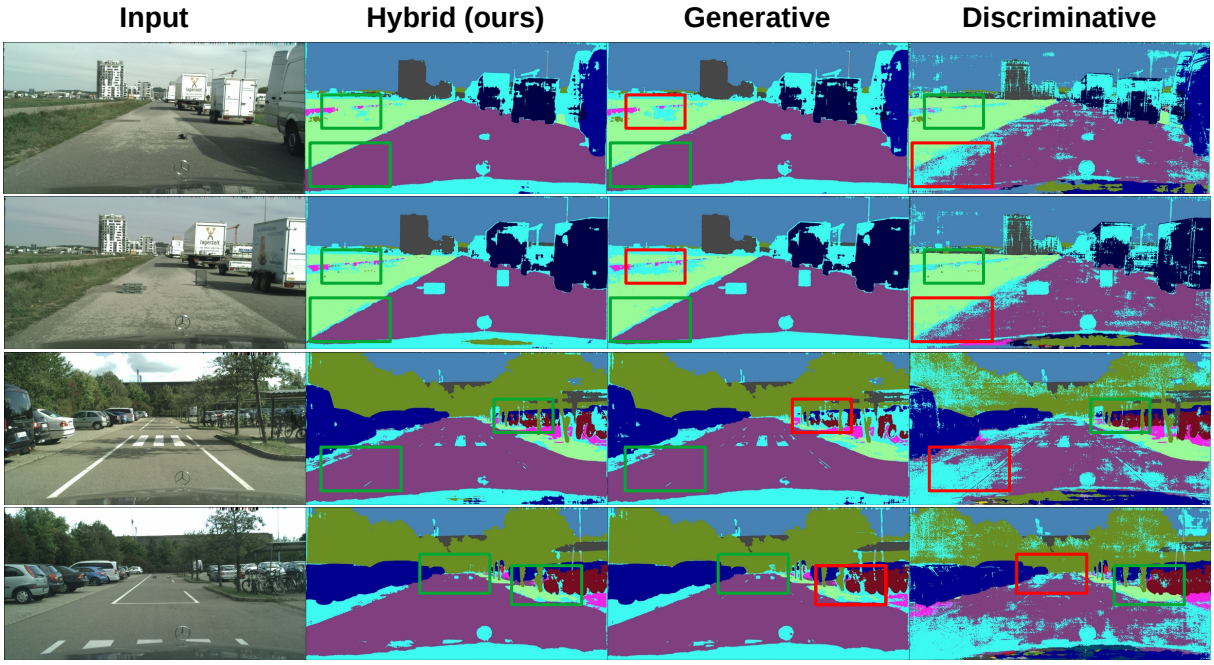


**Figure 6.7:** Open-set segmentation on FS LostAndFound with discriminative, generative and hybrid anomaly detection. Hybrid anomaly detection yields the lowest FPR95 metric. Red and green boxes indicate the abundance and absence of false positive anomalies, respectively.

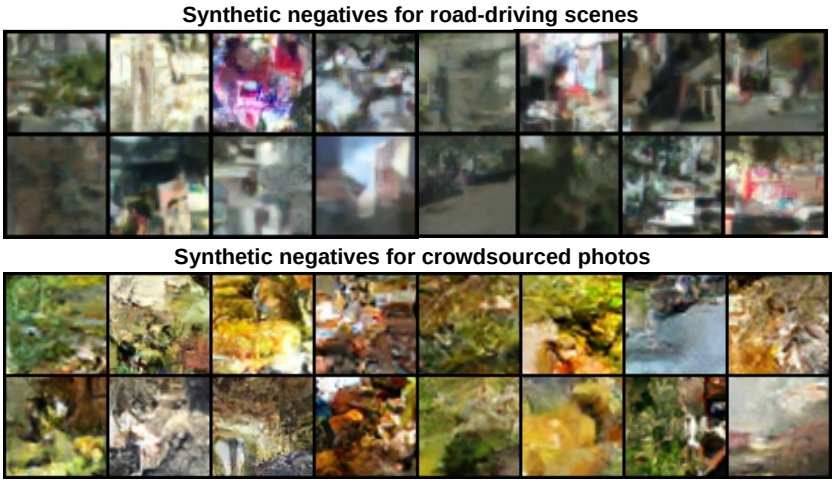


**Figure 6.8:** Dataset-specific synthetic negatives sampled from our normalizing flows (cf. Section 4.4). During the training, we sample the normalizing flow at different resolutions to mimic anomalies of different sizes.

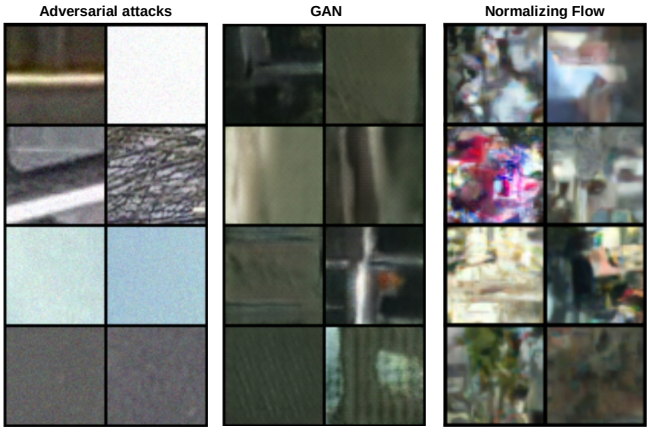


**Figure 6.9:** Comparison of synthetic negatives produced by local adversarial attacks, jointly trained GAN, and the proposed jointly trained normalizing flow. Our method produces synthetic negatives that diverge from the inliers while being more visually diverse than the other two approaches.

# Chapter 7

# Conclusion and outlook

## 7.1 Conclusion

This thesis proposed a novel approach for open-set inference that complements standard semantic segmentation models with dense semantic anomaly detector. The proposed approach, named *DenseHybrid*, strives for synergy between generative and discriminative anomaly detection. The generative anomaly detector is implemented atop the dense classifier by reinterpreting exponentiated logits as an unnormalized joint distribution of input and label. In this framework, data likelihood can be recovered by marginalization. The discriminative anomaly detector corresponds to the inlier posterior implemented by an additional binary classification head appended to the classifier.

A dense classifier equipped with DenseHybrid necessitates fine-tuning to attain adequate open-set competence. The proposed fine-tuning procedure eschews the evaluation of the intractable normalization constant introduced with unnormalized likelihood by leveraging negative training data. The negative data can be sourced from a general-purpose dataset, generated by a jointly trained normalizing flow, or sampled as a mixture of both sources. To produce high-quality synthetic negatives, we resort to densely connected normalizing flow with stochastic skip connections named *DenseFlow*. The introduction of stochastic skip connections improves the generative modeling performance of normalizing flows on standard low-resolution image datasets.

The proposed DenseHybrid is evaluated on dense semantic anomaly detection and open-set segmentation. Evaluation protocols use the standard benchmarks and datasets that involve general-purpose scenes and application-specific road driving scenarios. The obtained performance is quantified with the standard evaluation metrics. Furthermore, open-set segmentation performance is measured with a novel metric that enables quantification of the performance gap between closed-set and open-set setups.

Experimental evaluation of DenseHybrid reveals consistent performance gains over alter-

native methods across multiple test scenarios. These performance gains come with neglectable computational overhead and minimal elongation of inference time. The strong performance of DenseHybrid is due to moderate correlation between generative and discriminative anomaly detectors. Thus, ensembling the two components forms a competetive hybrid anomaly detector.

## 7.2 Outlook

The conducted performance evaluation reveals a large gap between open and closed set segmentation performance. Closing this gap requires immediate attention to ensure safer deployments of deep models in the real world. Furthermore, interesting directions for future work could extend DenseHybrid towards other dense prediction tasks such as open-set panoptic segmentation.

Recent advances in generative models drastically improved the quality of generated images through large-scale training. Careful adaptations of these models may offer alternative sources for synthetic negative samples.

Finally, applying DenseHybrid for contemporary semantic segmentation approaches based on mask-level recognition may further close the gap between open and closed set performance.

# Chapter 8

# Appendix

## 8.1 DenseFlow data likelihood lower bound

Let $\mathbf{z}_i$ denote the input, which we consider to be distributed according to $p(\mathbf{z}_i)$. Let $\mathbf{e}_i$ be noise independent of $\mathbf{z}_i$ with a known distribution $p(\mathbf{e}_i)$. Let $p(\mathbf{h}_i)$ be a Gaussian distribution and $f$ a function representing a normalizing flow: $\mathbf{h}_i = f(\mathbf{z}_i, \mathbf{e}_i)$. The normalizing flow distribution

$$p(\mathbf{z}_i, \mathbf{e}_i) = p(\mathbf{h}_i) \left| \det \frac{\partial \mathbf{h}_i}{\partial (\mathbf{z}_i, \mathbf{e}_i)} \right| . \tag{8.1}$$

We do not have a guarantee that $p(\mathbf{z}_i) = p(\mathbf{z}_i, \mathbf{e}_i)/p(\mathbf{e}_i)$.

To get the density $p(\mathbf{z}_i)$, we have to marginalize $p(\mathbf{z}_i, \mathbf{e}_i)$:

$$p(\mathbf{z}_i) = \int p(\mathbf{z}_i, \mathbf{e}_i)\, d\mathbf{e}_i \,. \tag{8.2}$$

We can efficiently estimate the integral using importance sampling:

$$p(\mathbf{z}_i) = \int \frac{p(\mathbf{z}_i, \mathbf{e}_i)}{p(\mathbf{e}_i)} p(\mathbf{e}_i)\, d\mathbf{e}_i = \mathbb{E}_{\mathbf{e}_i \sim p(\mathbf{e}_i)} \left[ \frac{p(\mathbf{z}_i, \mathbf{e}_i)}{p(\mathbf{e}_i)} \right] . \tag{8.3}$$

Log-likelihood can be computed as:

$$\ln p(\mathbf{z}_i) = \ln \mathbb{E}_{\mathbf{e}_i \sim p(\mathbf{e}_i)} \left[ \frac{p(\mathbf{z}_i, \mathbf{e}_i)}{p(\mathbf{e}_i)} \right] . \tag{8.4}$$

By applying Jensen's inequality, we obtain a lower bound on the log-likelihood,

$$\ln p(\mathbf{z}_i) \geq \mathbb{E}_{\mathbf{e}_i \sim p(\mathbf{e}_i)} \left[ \ln p(\mathbf{z}_i, \mathbf{e}_i) - \ln p(\mathbf{e}_i) \right] , \tag{8.5}$$

which corresponds to Equation (3.16).

## 8.2 Toy example dataset

Here we explain the data generation process and model architecture used in 2D toy example from Section 4.2. Inlier datapoints are generated by sampling the Gaussian mixture:

$$p_{\text{in}}(\mathbf{x}) = 0.5 \cdot \mathcal{N}(\mu_1, \Sigma_1) + 0.5 \cdot \mathcal{N}(\mu_2, \Sigma_2), \tag{8.6}$$

where $\mu_1 = \mu_2 = 0$ while $\Sigma_1 = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.1 \end{bmatrix}$ and $\Sigma_2 = \begin{bmatrix} 0.071 & 0.071 \\ -0.639 & 0.639 \end{bmatrix}$. The majority of negative training data is located in the first and fourth quadrants of the considered space in order to imitate a finite negative dataset. Outlier test data encompass the inlier distribution.

The discriminative anomaly detector is a binary classifier which consists of 4 MLP layers and ReLU activations. The generative anomaly detector is an energy-based model with a similar architecture as the binary classifier. The hybrid anomaly score combines the generative and the discriminative scores as proposed in our method. We visualize all three anomaly scores on the same scale. This can be done since the induced rankings are invariant to monotonic transformations. To ensure reproducibility, all samples are generated with the random seed set to 7. Different seeds also yield similar results.

## 8.3 On effectiveness of hybrid anomaly detector

Let us consider anomaly scoring function $s : \mathscr{X} \to \mathbb{R}$ which assigns higher values to anomalies and lower values to normal data. Without loss of generality, we can assume that the assigned scores are standardized (they have zero mean and unit variance) since ranking functions are invariant to scaling with positive values and shifting, which are required for standardization. Let $f : \mathscr{X} \to \{-1, +1\}$ be a labeling function which outputs +1 if a given input is an anomaly and -1 otherwise. We can decompose the anomaly scoring function $s$ into a correct labeling $f$ and an error function $\varepsilon$:

$$s(\mathbf{x}) = f(\mathbf{x}) + \varepsilon(\mathbf{x}). \tag{8.7}$$

We can compute the expected squared error of a scoring function $s$ as:

$$\mathscr{E}(s) = \mathbb{E}_{\mathbf{x}}[(s(\mathbf{x}) - f(\mathbf{x}))^2] = \mathbb{E}_{\mathbf{x}}[(\varepsilon(\mathbf{x}))^2]. \tag{8.8}$$

Note that $\mathscr{E}(s) = 0$ implies perfect separation between inliers and outliers, and therefore leads to perfect score in terms of AP, AUROC and FPR at $\text{TPR}_{95}$.

Our goal is now to show conditions under which the expected square error of hybrid anomaly

detector is lower than of the expected error of the best component:

$$\mathscr{E}(s_H) < \inf\{\mathscr{E}(s_G), \mathscr{E}(s_D)\}. \tag{8.9}$$

The anomaly score $s_G$ is a function of data likelihood and therefore generative anomaly detector. The anomaly score $s_D$ is a function of the inlier posterior, that is discriminative anomaly detector. We can define a hybrid anomaly score $s_H$ as an average of the two components:

$$s_H(\mathbf{x}) := \frac{1}{2}s_D(\mathbf{x}) + \frac{1}{2}s_G(\mathbf{x}). \tag{8.10}$$

Then, the expected squared error of the hybrid anomaly score $s_H$ equals:

$$\begin{aligned}
\mathscr{E}(s_H) &= \mathbb{E}_{\mathbf{x}}[(s_H(\mathbf{x}) - f(\mathbf{x}))^2] \\
&= \mathbb{E}_{\mathbf{x}}\left[\left(\frac{1}{2}\varepsilon_D(\mathbf{x}) + \frac{1}{2}\varepsilon_G(\mathbf{x}) + f(\mathbf{x}) - f(\mathbf{x})\right)^2\right] \\
&= \mathbb{E}_{\mathbf{x}}\left[\left(\frac{1}{2}\varepsilon_D(\mathbf{x}) + \frac{1}{2}\varepsilon_G(\mathbf{x})\right)^2\right] \\
&= \frac{1}{4}\mathbb{E}_{\mathbf{x}}\left[(\varepsilon_D(\mathbf{x}))^2 + (\varepsilon_G(\mathbf{x}))^2\right] + \frac{1}{2}\mathbb{E}_{\mathbf{x}}[\varepsilon_D(\mathbf{x})\varepsilon_G(\mathbf{x})] \\
&= \frac{1}{4}\mathscr{E}(s_D) + \frac{1}{4}\mathscr{E}(s_G) + \frac{1}{2}\mathrm{cov}(\varepsilon_D, \varepsilon_G) \\
&+ \frac{1}{2}\mathbb{E}_{\mathbf{x}}[\varepsilon_D(\mathbf{x})] \cdot \mathbb{E}_{\mathbf{x}}[\varepsilon_G(\mathbf{x})] \\
&= \frac{1}{4}\mathscr{E}(s_D) + \frac{1}{4}\mathscr{E}(s_G) + C_1\rho(\varepsilon_D, \varepsilon_G) + C_2
\end{aligned} \tag{8.11}$$

Note that $\rho$ represents Pearson's correlation coefficient, while $C_2 = \frac{1}{2} \cdot \mathbb{E}_{\mathbf{x}}[\varepsilon_D(\mathbf{x})] \cdot \mathbb{E}_{\mathbf{x}}[\varepsilon_G(\mathbf{x})]$ and $C_1 = \frac{1}{2} \cdot \mathrm{std}(\varepsilon_D) \cdot \mathrm{std}(\varepsilon_G)$. Therefore, by joining (8.11) and (8.9) our goal becomes equivalent to the following inequality:

$$\frac{1}{4}\mathscr{E}(s_D) + \frac{1}{4}\mathscr{E}(s_G) + C_1\rho(\varepsilon_D, \varepsilon_G) + C_2 < \min\{\mathscr{E}(s_D), \mathscr{E}(s_G)\}. \tag{8.12}$$

Without loss of generality, we can assume $\mathscr{E}(s_G) < \mathscr{E}(s_D)$. Then, we denote $\mathscr{E}(s_G) = e$ and $\mathscr{E}(s_D) = \alpha \cdot e, \alpha > 1$. We can now rewrite (8.12) as:

$$\frac{\alpha - 3}{4}e + C_1\rho(\varepsilon_D, \varepsilon_G) + C_2 < 0\,. \tag{8.13}$$

We can see that the effectiveness of our hybrid anomaly detector depends on the ratio between the errors of the two components $\alpha = \frac{\max\{\mathscr{E}(s_G), \mathscr{E}(s_D)\}}{\min\{\mathscr{E}(s_G), \mathscr{E}(s_D)\}}$, their correlation $\rho$, the level of error $e = \min\{\mathscr{E}(s_G), \mathscr{E}(s_D)\}$, and constants $C_1, C_2$. Consequently, equation (8.13) provides a sufficient condition that the hybrid anomaly detector must satisfy to be effective. Our hybrid anomaly

detector indeed satisfies these conditions in a practical setting (cf. Sec. 6.5).

Finally, we have to show that our hybrid anomaly detector can be viewed as an ensemble over $s_G$ and $s_D$:

$$
\begin{aligned}
s_H(\mathbf{x}) &= \frac{1}{2}s_D(\mathbf{x}) + \frac{1}{2}s_G(\mathbf{x}) && (8.14)\\
&= \frac{1}{2}\frac{s'_D(\mathbf{x}) - \mu_D}{\sigma_D} + \frac{1}{2}\frac{s'_G(\mathbf{x}) - \mu_G}{\sigma_G} && (8.15)\\
&= \frac{1}{2\sigma_D}s'_D(\mathbf{x}) + \frac{1}{2\sigma_G}s'_G(\mathbf{x}) - \left(\frac{\mu_D}{2\sigma_D} + \frac{\mu_G}{2\sigma_G}\right) && (8.16)\\
&= \frac{1}{2\sigma}(s'_D(\mathbf{x}) + s'_G(\mathbf{x})) - \left(\frac{\mu_D}{2\sigma_D} + \frac{\mu_G}{2\sigma_G}\right) && (8.17)\\
&= A \cdot (\underbrace{(\ln P(d_{\text{out}}|\mathbf{x}))}_{s'_D(\mathbf{x})} + \underbrace{(-\ln p(\mathbf{x}))}_{s'_G(\mathbf{x})}) + C && (8.18)\\
&= A \cdot (\underbrace{(\ln P(d_{\text{out}}|\mathbf{x}))}_{s'_D(\mathbf{x})} + \underbrace{(-\ln \hat{p}(\mathbf{x})) + \ln Z)}_{s'_G(\mathbf{x})} + C && (8.19)\\
&\cong \ln P(d_{\text{out}}|\mathbf{x}) - \ln \hat{p}(\mathbf{x}). && (8.20)
\end{aligned}
$$

Note that we have assumed $\sigma = \sigma_D = \sigma_G$.

## 8.4 DenseHybrid data likelihood objective

We present a step-by-step derivation of Equation (4.13) as follows. Note that normalization constant Z cancels out, while LSE denotes log-sum-exp.

$$
\begin{aligned}
L_{\mathbf{x}}(\theta) &= \mathbb{E}_{\mathbf{x}\in D_{\text{in}}}[-\ln p_\theta(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\in D_{\text{out}}}[-\ln p_\theta(\mathbf{x})]\\
&= \mathbb{E}_{\mathbf{x}\in D_{\text{in}}}[-\ln \hat{p}_\theta(\mathbf{x}) + \ln Z(\theta)] - \mathbb{E}_{\mathbf{x}\in D_{\text{out}}}[-\ln \hat{p}_\theta(\mathbf{x}) + \ln Z(\theta)]\\
&= \mathbb{E}_{\mathbf{x}\in D_{\text{in}}}[-\ln \hat{p}_\theta(\mathbf{x})] + \cancel{\ln Z(\theta)} - \mathbb{E}_{\mathbf{x}\in D_{\text{out}}}[-\ln \hat{p}_\theta(\mathbf{x})] - \cancel{\ln Z(\theta)}\\
&= \mathbb{E}_{\mathbf{x}\in D_{\text{in}}}[-\ln \hat{p}_\theta(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\in D_{\text{out}}}[-\ln \hat{p}_\theta(\mathbf{x})]\\
&= -\mathbb{E}_{D_{\text{in}}}[\operatorname*{LSE}_i(\mathbf{s}_i)] + \mathbb{E}_{D_{\text{out}}}[\operatorname*{LSE}_i(\mathbf{s}_i)] \qquad \mathbf{s} = f_\theta(\mathbf{x})\\
&\leq -\mathbb{E}_{\mathbf{x},y\in D_{\text{in}}}[(\mathbf{s}_y)] + \mathbb{E}_{D_{\text{out}}}[\operatorname*{LSE}_i(\mathbf{s}_i)] = L_{\mathbf{x}}^{\text{UB}}(\theta) && (8.21)
\end{aligned}
$$

The last inequality holds since the log-sum-exp operator over a set of elements is always greater than the individual elements.

We can connect the gradient of $L_{\mathbf{x}}$ w.r.t parameters with the gradient of negative log-likelihood:

$$\begin{aligned}
\nabla_\theta \mathbb{E}_{\mathbf{x}\sim p_D}[-\ln p_\theta(\mathbf{x})] &= -\mathbb{E}_{\mathbf{x}\sim p_D}[\nabla_\theta \ln \hat{p}_\theta(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_\theta}[\nabla_\theta \ln \hat{p}_\theta(\mathbf{x})] \\
&= -\mathbb{E}_{\mathbf{x}\sim p_D}[\nabla_\theta \ln \hat{p}_\theta(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim q}\left[\frac{p_\theta(\mathbf{x})}{q(\mathbf{x})}\nabla_\theta \ln \hat{p}_\theta(\mathbf{x})\right] \\
&= -\mathbb{E}_{\mathbf{x}\sim p_D}[\nabla_\theta \ln \hat{p}_\theta(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim q}[\nabla_\theta \ln \hat{p}_\theta(\mathbf{x})] + b(\mathbf{x}) \\
&\approx -\mathbb{E}_{\mathbf{x}\sim p_D}[\nabla_\theta \ln \hat{p}_\theta(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim q}[\nabla_\theta \ln \hat{p}_\theta(\mathbf{x})] \\
&\approx -\mathbb{E}_{\mathbf{x}\in D_{\text{in}}}[\nabla_\theta \ln \hat{p}_\theta(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\in D_{\text{out}}}[\nabla_\theta \ln \hat{p}_\theta(\mathbf{x})] = \nabla_\theta L_{\mathbf{x}}(\theta) \qquad (8.22)
\end{aligned}$$

Here, we replace slow sampling of $p_\theta$ with proposal distribution $q$ that has a similar support set and can be efficiently sampled. Consequently, the expression (8.22) can be quickly evaluated but results in a biased gradient since $b(\mathbf{x}) = \mathbb{E}_{\mathbf{x}\sim q}\left[(\frac{p_\theta(\mathbf{x})}{q(\mathbf{x})} - 1)\nabla_\theta \ln \hat{p}_\theta(\mathbf{x})\right]$. In practice, our negative dataset (e.g. ADE20k) consists of both known and unknown classes which is sufficiently good approximation of $q$.

## 8.5 Compound DenseHybrid objective

We present a step-by-step derivation of Equation (4.16) as follows. Recall that the standard cross entropy $L_{\text{cls}}$ equals to:

$$L_{\text{cls}}(\theta) = \mathbb{E}_{\mathbf{x},y\in D_{\text{in}}}[-\ln p(y|\mathbf{x})] = -\mathbb{E}_{\mathbf{x},y\in D_{\text{in}}}[\mathbf{s}_y] + \mathbb{E}_{\mathbf{x},y\in D_{\text{in}}}[\operatorname*{LSE}_{y'}(\mathbf{s}_{y'})] \qquad (8.23)$$

The upper bound to data likelihood $L_{\mathbf{cls}}^{\text{UB}}$ equals to:

$$L_{\mathbf{x}}^{\text{UB}}(\theta) = -\mathbb{E}_{\mathbf{x},y\in D_{\text{in}}}[\mathbf{s}_y] + \mathbb{E}_{\mathbf{x}\in D_{\text{out}}}[\operatorname*{LSE}_{i}(\mathbf{s}_i)]. \qquad (8.24)$$

These two losses have a term in common. Consequently, we can omit one of them in the joint loss:

$$L_{\text{cls}}(\theta) + L_{\mathbf{x}}^{\text{UB}}(\theta) = -2\,\mathbb{E}_{\mathbf{x},y\in D_{\text{in}}}[\mathbf{s}_y] + \mathbb{E}_{\mathbf{x},y\in D_{\text{in}}}[\operatorname*{LSE}_{y'}(\mathbf{s}_{y'})] + \mathbb{E}_{\mathbf{x}\in D_{\text{out}}}[\operatorname*{LSE}_{i}(\mathbf{s}_i)]. \qquad (8.25)$$

By omitting the multiplicative constant, the above expression becomes:

$$L_{\text{cls}}(\theta) + L_{\mathbf{x}}^{\text{UB}}(\theta) = -\mathbb{E}_{\mathbf{x},y\in D_{\text{in}}}[\ln P(y|\mathbf{x})] - \mathbb{E}_{\mathbf{x}\in D_{\text{out}}}[\ln \hat{p}(\mathbf{x})] \qquad (8.26)$$

By further adding the data posterior loss and grouping terms according to the expectations

we obtain:

$$\begin{aligned} L(\theta,\gamma) &= L_{\text{cls}}(\theta) + L_{\mathbf{x}}^{\text{UB}}(\theta) + L_{\mathbf{d}}(\theta,\gamma) \\ &= -\mathbb{E}_{\mathbf{x},y\in D_{\text{in}}}[\ln P(y|\mathbf{x}) + \ln P(d_{\text{in}}|\mathbf{x})] - \mathbb{E}_{\mathbf{x}\in D_{\text{out}}}[\ln(1 - P(d_{\text{in}}|\mathbf{x})) - \ln \hat{p}(\mathbf{x})]. \end{aligned}$$

In practice, we introduce loss modulation hyperparameters which control the impact of each loss term.

## 8.6 Extended DenseFlow results

Table 8.1 compares DenseFlow results in density estimation with other families of generative models.
Table 8.2 compares DenseFow results in image generation quality with other families of generative models.

**Table 8.1:** Likelihood evaluation (in bits/dim) on standard datasets.

| Model type | Method | CIFAR-10 32x32 | ImageNet 32x32 | CelebA 64x64 | ImageNet 64x64 |
|---|---|---|---|---|---|
| Variational Autoencoders | Conv Draw [134] | 3.58 | 4.40 | - | 4.10 |
| | DVAE++ [135] | 3.38 | - | - | - |
| | IAF-VAE [136] | 3.11 | - | - | - |
| | BIVA [137] | 3.08 | 3.96 | 2.48 | - |
| | CR-NVAE [138] | **2.51** | - | **1.86** | - |
| Diffusion models | DDPM [139] | 3.70 | - | - | - |
| | UDM (RVE) + ST [140] | 3.04 | - | 1.93 | - |
| | Imp. DDPM [141] | 2.94 | - | - | 3.53 |
| | VDM [142] | 2.65 | 3.72 | - | 3.40 |
| Autoregressive Models | Gated PixelCNN [89] | 3.03 | 3.83 | - | 3.57 |
| | PixelRNN [143] | 3.00 | 3.86 | - | 3.63 |
| | PixelCNN++ [144] | 2.92 | - | - | - |
| | Image Transformer [145] | 2.90 | 3.77 | 2.61 | - |
| | PixelSNAIL [146] | 2.85 | 3.80 | - | - |
| | SPN [147] | - | 3.85 | - | 3.53 |
| | Routing transformer [148] | 2.95 | - | - | 3.43 |
| Normalizing Flows | Real NVP [98] | 3.49 | 4.28 | 3.02 | 3.98 |
| | GLOW [99] | 3.35 | 4.09 | - | 3.81 |
| | Wavelet Flow [125] | - | 4.08 | - | 3.78 |
| | Residual Flow [102] | 3.28 | 4.01 | - | 3.78 |
| | i-DenseNet [126] | 3.25 | 3.98 | - | - |
| | Flow++ [100] | 3.08 | 3.86 | - | 3.69 |
| | ANF [108] | 3.05 | 3.92 | - | 3.66 |
| | VFlow [107] | 2.98 | 3.83 | - | 3.66 |
| Hybrid Architectures | mAR-SCF [149] | 3.22 | 3.99 | - | 3.80 |
| | MaCow [150] | 3.16 | - | - | 3.69 |
| | SurVAE Flow [151] | 3.08 | 4.00 | - | 3.70 |
| | NVAE [152] | 2.91 | 3.92 | 2.03 | - |
| | PixelVAE++ [153] | 2.90 | - | - | - |
| | $\delta$-VAE [154] | 2.83 | 3.77 | - | - |
| | DenseFlow-74-10 (ours) | 2.98 | **3.63** | 1.99 | **3.35** |

**Table 8.2:** Evaluation of FID score on CIFAR-10.

| **Model type** | **Model** | **FID** ↓ |
|---|---|---|
| Autoregressive | PixelCNN [143, 155] | 65.93 |
| Models | PixelIQN [155] | 49.46 |
| | i-ResNet [103] | 65.01 |
| Normalizing | Glow [99] | 46.90 |
| Flows | Residual flow [102] | 46.37 |
| | ANF [108] | 30.60 |
| | DCGAN [155, 156] | 37.11 |
| GANs | WGAN-GP [155, 157] | 36.40 |
| | DA-StyleGAN V2 [158] | 5.79 |
| | VDM [142] | 4.00 |
| Diffusion models | DDPM [139] | 3.17 |
| | UDM (RVE) + ST [140] | 2.33 |
| | SurVAE-flow [151] | 49.03 |
| Hybrid | mAR-SCF [149] | 33.06 |
| Architectures | VAEBM [159] | 12.19 |
| | DenseFlow-74-10 (ours) | 34.90 |

# Bibliography

[1] Bauer, J., Baumli, K., Behbahani, F. M. P., Bhoopchand, A., Bradley-Schmieg, N., Chang, M., Clay, N., Collister, A., Dasagi, V., Gonzalez, L., Gregor, K., Hughes, E., Kashem, S., Loks-Thompson, M., Openshaw, H., Parker-Holder, J., Pathak, S., Nieves, N. P., Rakicevic, N., Rocktäschel, T., Schroecker, Y., Singh, S., Sygnowski, J., Tuyls, K., York, S., Zacherl, A., Zhang, L. M., "Human-timescale adaptation in an open-ended task space", in International Conference on Machine Learning. PMLR, 2023.

[2] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., Hassabis, D., "Highly accurate protein structure prediction with alphafold", Nature, Vol. 596, 2021.

[3] Fan, S., Ponisio, M. R., Xiao, P., Ha, S. M., Chakrabarty, S., Lee, J. J., Flores, S., LaMontagne, P., Gordon, B., Raji, C. A., Marcus, D. S., Nazeri, A., Ances, B. M., Bateman, R. J., Morris, J. C., Benzinger, T. L. S., Sotiras, A., Atzen, S., "Amyloidpetnet: Classification of amyloid positivity in brain pet imaging using end-to-end deep learning", Radiology, Vol. 311, No. 3, 2024.

[4] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., "The cityscapes dataset for semantic urban scene understanding", in IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016.

[5] Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., Lu, L., Jia, X., Liu, Q., Dai, J., Qiao, Y., Li, H., "Planning-oriented autonomous driving", in IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2023.

[6] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L., "ImageNet Large Scale

Visual Recognition Challenge", International Journal of Computer Vision (IJCV), Vol. 115, No. 3, 2015, str. 211-252.

[7] Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A., "Semantic understanding of scenes through the ADE20K dataset", Int. J. Comput. Vis., Vol. 127, No. 3, 2019, str. 302–321.

[8] Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., Müller, K., "A unifying review of deep and shallow anomaly detection", Proc. IEEE, 2021.

[9] Blum, H., Sarlin, P.-E., Nieto, J., Siegwart, R., Cadena, C., "The fishyscapes benchmark: Measuring blind spots in semantic segmentation", International Journal of Computer Vision, Vol. 129, 2021.

[10] Bevandić, P., Krešo, I., Oršić, M., Šegvić, S., "Dense open-set recognition based on training with noisy negative images", Image and Vision Computing, Vol. 124, 2022, str. 104490.

[11] Chan, R., Rottmann, M., Gottschalk, H., "Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation", in International Conference on Computer Vision, ICCV, 2021.

[12] Bevandic, P., Kreso, I., Orsic, M., Segvic, S., "Simultaneous semantic segmentation and outlier detection in presence of domain shift", in 41st DAGM German Conference, DAGM GCPR, 2019.

[13] Blum, H., Sarlin, P., Nieto, J. I., Siegwart, R., Cadena, C., "Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving", in 2019 IEEE/CVF International Conference on Computer Vision Workshops. IEEE, 2019, str. 2403–2412.

[14] Liang, C., Wang, W., Miao, J., Yang, Y., "Gmmseg: Gaussian mixture based generative semantic segmentation models", Advances in Neural Information Processing Systems, 2022.

[15] Hendrycks, D., Basart, S., Mazeika, M., Zou, A., Kwon, J., Mostajabi, M., Steinhardt, J., Song, D., "Scaling out-of-distribution detection for real-world settings", in International Conference on Machine Learning, ICML, 2022.

[16] Grcic, M., Bevandic, P., Segvic, S., "Densehybrid: Hybrid anomaly detection for dense open-set recognition", in European Conference on Computer Vision, ECCV, 2022.

[17] Grathwohl, W., Wang, K., Jacobsen, J., Duvenaud, D., Norouzi, M., Swersky, K., "Your classifier is secretly an energy based model and you should treat it like one", in International Conference on Learning Representations, ICLR, 2020.

[18] Grcić, M., Šegvić, S., "Hybrid open-set segmentation with synthetic negative data", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

[19] Grcić, M., Grubišić, I., Šegvić, S., "Densely connected normalizing flows", in Neural Information Processing Systems, 2021.

[20] Hawkins, D. M., Identification of Outliers, ser. Monographs on Applied Probability and Statistics. Springer, 1980.

[21] Han, S., Hu, X., Huang, H., Jiang, M., Zhao, Y., "Adbench: Anomaly detection benchmark", in Neural Information Processing Systems, 2022.

[22] Yang, J., Wang, P., Zou, D., Zhou, Z., Ding, K., Peng, W., Wang, H., Chen, G., Li, B., Sun, Y., Du, X., Zhou, K., Zhang, W., Hendrycks, D., Li, Y., Liu, Z., "Openood: Benchmarking generalized out-of-distribution detection", in Neural Information Processing Systems 2022, 2022.

[23] Chandola, V., Banerjee, A., Kumar, V., "Anomaly detection: A survey", ACM Comput. Surv., Vol. 41, No. 3, 2009, str. 15:1–15:58.

[24] Hendrycks, D., Gimpel, K., "A baseline for detecting misclassified and out-of-distribution examples in neural networks", in 5th International Conference on Learning Representations, ICLR, 2017.

[25] Liang, S., Li, Y., Srikant, R., "Enhancing the reliability of out-of-distribution image detection in neural networks", in 6th International Conference on Learning Representations, ICLR, 2018.

[26] Kendall, A., Gal, Y., "What uncertainties do we need in bayesian deep learning for computer vision?", in Neural Information Processing Systems, 2017.

[27] Lakshminarayanan, B., Pritzel, A., Blundell, C., "Simple and scalable predictive uncertainty estimation using deep ensembles", in Neural Information Processing Systems, 2017.

[28] Huang, R., Geng, A., Li, Y., "On the importance of gradients for detecting distributional shifts in the wild", in Neural Information Processing Systems, 2021, str. 677–689.

[29] Oberdiek, P., Rottmann, M., Gottschalk, H., "Classification uncertainty of deep neural networks based on gradient information", in Artificial Neural Networks in Pattern Recognition, 2018.

[30] Dhamija, A. R., Günther, M., Boult, T. E., "Reducing network agnostophobia", in Annual Conference on Neural Information Processing Systems 2018, NeurIPS, 2018.

[31] Hendrycks, D., Mazeika, M., Dietterich, T. G., "Deep anomaly detection with outlier exposure", in 7th International Conference on Learning Representations, ICLR, 2019.

[32] Liu, W., Wang, X., Owens, J. D., Li, Y., "Energy-based out-of-distribution detection", in NeurIPS, 2020.

[33] Chen, J., Li, Y., Wu, X., Liang, Y., Jha, S., "ATOM: robustifying out-of-distribution detection using outlier mining", in Machine Learning and Knowledge Discovery in Databases, Vol. 12977. Springer, 2021, str. 430–445.

[34] Ming, Y., Fan, Y., Li, Y., "POEM: out-of-distribution detection with posterior sampling", in International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA, ser. Proceedings of Machine Learning Research, Vol. 162. PMLR, 2022, str. 15 650–15 665.

[35] Jiang, W., Cheng, H., Chen, M., Wang, C., Wei, H., "DOS: Diverse outlier sampling for out-of-distribution detection", in International Conference on Learning Representations, 2024.

[36] Wang, Q., Fang, Z., Zhang, Y., Liu, F., Li, Y., Han, B., "Learning to augment distributions for out-of-distribution detection", in Neural Information Processing Systems, 2023.

[37] Sun, Y., Guo, C., Li, Y., "React: Out-of-distribution detection with rectified activations", in NeurIPS, 2021.

[38] Djurisic, A., Bozanic, N., Ashok, A., Liu, R., "Extremely simple activation shaping for out-of-distribution detection", in International Conference on Learning Representations, 2023.

[39] Sun, Y., Li, Y., "DICE: leveraging sparsification for out-of-distribution detection", in European Conference on Computer Vision. Springer, 2022, str. 691–708.

[40] Sun, Y., Ming, Y., Zhu, X., Li, Y., "Out-of-distribution detection with deep nearest neighbors", in International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, Vol. 162. PMLR, 2022, str. 20 827–20 840.

[41] Nalisnick, E. T., Matsukawa, A., Teh, Y. W., Görür, D., Lakshminarayanan, B., "Do deep generative models know what they don't know?", in International Conference on Learning Representations, 2019.

[42] Serrà, J., Álvarez, D., Gómez, V., Slizovskaia, O., Núñez, J. F., Luque, J., "Input complexity and out-of-distribution detection with likelihood-based generative models", in 8th International Conference on Learning Representations, ICLR, 2020.

[43] Zhang, L. H., Goldstein, M., Ranganath, R., "Understanding failures in out-of-distribution detection with deep generative models", in International Conference on Machine Learning, ICML, 2021.

[44] Du, X., Wang, Z., Cai, M., Li, Y., "VOS: learning what you don't know by virtual outlier synthesis", in The Tenth International Conference on Learning Representations, ICLR 2022, 2022.

[45] Zhang, H., Li, A., Guo, J., Guo, Y., "Hybrid models for open set recognition", in European Conference on Computer Vision, 2020.

[46] Jung, S., Lee, J., Gwak, D., Choi, S., Choo, J., "Standardized max logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation", in International Conference on Computer Vision, ICCV, 2021.

[47] Malinin, A., Gales, M. J. F., "Predictive uncertainty estimation via prior networks", in Neural Information Processing Systems, 2018.

[48] Lucas, T., Shmelkov, K., Alahari, K., Schmid, C., Verbeek, J., "Adaptive density estimation for generative models", in Neural Information Processing Systems, 2019.

[49] Tian, Y., Liu, Y., Pang, G., Liu, F., Chen, Y., Carneiro, G., "Pixel-wise energy-biased abstention learning for anomaly segmentation on complex urban driving scenes", in European Conference on Computer Vision, 2022.

[50] Lis, K., Nakka, K. K., Fua, P., Salzmann, M., "Detecting the unexpected via image resynthesis", in International Conference on Computer Vision, ICCV, 2019.

[51] Xia, Y., Zhang, Y., Liu, F., Shen, W., Yuille, A. L., "Synthesize then compare: Detecting failures and anomalies for semantic segmentation", in European Conference on Computer Vision, ECCV, 2020.

[52] Biase, G. D., Blum, H., Siegwart, R., Cadena, C., "Pixel-wise anomaly detection in complex driving scenes", in Computer Vision and Pattern Recognition, CVPR, 2021.

[53] Vojir, T., Šipka, T., Aljundi, R., Chumerin, N., Reino, D. O., Matas, J., "Road anomaly detection by partial image reconstruction with segmentation coupling", in International Conference on Computer Vision, ICCV, 2021.

[54] Fu, Y., Gao, D., Liu, T., Zheng, H., Hao, D., Pan, Z., "Evolving into a transformer: From a training-free retrieval-based method for anomaly obstacle segmentation", IEEE Transactions on Image Processing, Vol. 32, 2023, str. 6195–6209, dostupno na: http://dx.doi.org/10.1109/TIP.2023.3312910

[55] Zhao, Z., Cao, L., Lin, K., "Revealing the distributional vulnerability of discriminators by implicit generators", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 45, No. 7, 2023, str. 8888–8901.

[56] Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., Boult, T. E., "Toward open set recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, No. 7, 2013, str. 1757-1772.

[57] Scheirer, W. J., Jain, L. P., Boult, T. E., "Probability models for open set recognition", IEEE Trans. Pattern Anal. Mach. Intell., 2014.

[58] Bendale, A., Boult, T. E., "Towards open set deep networks", in IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[59] Cen, J., Yun, P., Cai, J., Wang, M. Y., Liu, M., "Deep metric learning for open world semantic segmentation", in International Conference on Computer Vision (ICCV), 2021.

[60] Chen, G., Qiao, L., Shi, Y., Peng, P., Li, J., Huang, T., Pu, S., Tian, Y., "Learning open set network with discriminative reciprocal points", in European Conference on Computer Vision, ser. Lecture Notes in Computer Science. Springer, 2020, str. 507–522.

[61] Chen, G., Peng, P., Wang, X., Tian, Y., "Adversarial reciprocal points learning for open set recognition", IEEE Trans. Pattern Anal. Mach. Intell., 2022.

[62] Vaze, S., Han, K., Vedaldi, A., Zisserman, A., "Open-set recognition: A good closed-set classifier is all you need", in The Tenth International Conference on Learning Representations, ICLR 2022, 2022.

[63] Boult, T. E., Cruz, S., Dhamija, A. R., Günther, M., Henrydoss, J., Scheirer, W. J., "Learning and the unknown: Surveying steps toward open world recognition", in AAAI Conference on Artificial Intelligence. AAAI Press, 2019.

[64] Geng, C., Huang, S., Chen, S., "Recent advances in open set recognition: A survey", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 43, No. 10, 2021, str. 3614–3631.

[65] Zendel, O., Honauer, K., Murschitz, M., Steininger, D., Dominguez, G. F., "Wilddash - creating hazard-aware benchmarks", in European Conference on Computer Vision (ECCV), 2018.

[66] Chan, R., Lis, K., Uhlemeyer, S., Blum, H., Honari, S., Siegwart, R., Fua, P., Salzmann, M., Rottmann, M., "Segmentmeifyoucan: A benchmark for anomaly segmentation", in Neural Information Processing Systems Track on Datasets and Benchmarks, 2021.

[67] Sokolova, M., Lapalme, G., "A systematic analysis of performance measures for classification tasks", Inf. Process. Manag., Vol. 45, No. 4, 2009, str. 427–437.

[68] Scherreik, M. D., Rigling, B. D., "Open set recognition for automatic target classification with rejection", IEEE Trans. Aerosp. Electron. Syst., Vol. 52, No. 2, 2016, str. 632–642.

[69] Sakaridis, C., Dai, D., Gool, L. V., "Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 44, No. 6, 2022.

[70] Neal, L., Olson, M. L., Fern, X. Z., Wong, W., Li, F., "Open set learning with counterfactual images", in European Conference on Computer Vision, 2018.

[71] Lee, K., Lee, H., Lee, K., Shin, J., "Training confidence-calibrated classifiers for detecting out-of-distribution samples", in 6th International Conference on Learning Representations, ICLR, 2018.

[72] Kong, S., Ramanan, D., "Opengan: Open-set recognition via open data generation", IEEE Trans. Pattern Anal. Mach. Intell., 2022.

[73] Grcić, M., Bevandić, P., Šegvić, S., "Dense open-set recognition with synthetic outliers generated by real NVP", in Int'l Conference on Computer Vision Theory and Applications, 2021.

[74] Grcic, M., Bevandic, P., Kalafatic, Z., Segvic, S., "Dense out-of-distribution detection by robust learning on synthetic negative data", Sensors, Vol. 24, No. 4, 2024, str. 1248.

[75] Kumar, N., Segvic, S., Eslami, A., Gumhold, S., "Normalizing flow based feature synthesis for outlier-aware object detection", in IEEE/CVF Computer Vision and Pattern Recognition, CVPR, 2023.

[76] Besnier, V., Bursuc, A., Picard, D., Briot, A., "Triggering failures: Out-of-distribution detection by learning from local adversarial attacks in semantic segmentation", in International Conference on Computer Vision, 2021.

[77] Hendrycks, D., Zou, A., Mazeika, M., Tang, L., Li, B., Song, D., Steinhardt, J., "Pixmix: Dreamlike pictures comprehensively improve safety measures", in IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2022, str. 16 762–16 771.

[78] Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., Lakshminarayanan, B., "Augmix: A simple data processing method to improve robustness and uncertainty", in International Conference on Learning Representations, ICLR, 2020.

[79] Michieli, U., Zanuttigh, P., "Knowledge distillation for incremental learning in semantic segmentation", Comput. Vis. Image Underst., Vol. 205, 2021, str. 103167.

[80] Uhlemeyer, S., Rottmann, M., Gottschalk, H., "Towards unsupervised open world semantic segmentation", in Uncertainty in Artificial Intelligence, 2022.

[81] Cao, K., Brbic, M., Leskovec, J., "Open-world semi-supervised learning", in International Conference on Learning Representations, ICLR, 2022.

[82] Grcic, M., Gadetsky, A., Brbic, M., "Fine-grained classes and how to find them", in International Conference on Machine Learning, 2024.

[83] Fu, Y., Wang, X., Dong, H., Jiang, Y., Wang, M., Xue, X., Sigal, L., "Vocabulary-informed zero-shot and open-set learning", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 42, No. 12, 2020.

[84] Romera-Paredes, B., Torr, P. H. S., "An embarrassingly simple approach to zero-shot learning", in International Conference on Machine Learning, ser. JMLR Workshop and Conference Proceedings, Vol. 37. JMLR.org, 2015, str. 2152–2161.

[85] Xian, Y., Lampert, C. H., Schiele, B., Akata, Z., "Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 41, No. 9, 2019.

[86] Salakhutdinov, R., Hinton, G., "Deep boltzmann machines", in Twelth International Conference on Artificial Intelligence and Statistics. PMLR, 2009.

[87] Du, Y., Mordatch, I., "Implicit generation and modeling with energy based models", in Neural Information Processing Systems 2019, NeurIPS 2019, 2019.

[88] Brooks, S., Gelman, A., Jones, G., Meng, X.-L., Handbook of Markov Chain Monte Carlo. Chapman and Hall/CRC, 2011.

[89] van den Oord, A., Kalchbrenner, N., Espeholt, L., Kavukcuoglu, K., Vinyals, O., Graves, A., "Conditional image generation with pixelcnn decoders", in Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, 2016, str. 4790–4798.

[90] Kingma, D. P., Welling, M., "Auto-encoding variational bayes", in 2nd International Conference on Learning Representations, ICLR, 2014.

[91] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., Ganguli, S., "Deep unsupervised learning using nonequilibrium thermodynamics", in Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, ser. JMLR Workshop and Conference Proceedings, Vol. 37. JMLR.org, 2015, str. 2256–2265.

[92] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., "Generative adversarial nets", in Neural Information Processing Systems, 2014.

[93] Bond-Taylor, S., Leach, A., Long, Y., Willcocks, C. G., "Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 44, No. 11, 2022, str. 7327-7347.

[94] Rezende, D. J., Mohamed, S., "Variational inference with normalizing flows", in International Conference on Machine Learning, ICML, 2015.

[95] Dinh, L., Krueger, D., Bengio, Y., "NICE: non-linear independent components estimation", in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings, 2015.

[96] Deisenroth, M. P., Faisal, A. A., Ong, C. S., Mathematics for Machine Learning. Cambridge University Press, 2020.

[97] Ardizzone, L., Lüth, C. T., Kruse, J., Rother, C., Köthe, U., "Guided image generation with conditional invertible neural networks", CoRR, Vol. abs/1907.02392, 2019, dostupno na: http://arxiv.org/abs/1907.02392

[98] Dinh, L., Sohl-Dickstein, J., Bengio, S., "Density estimation using real NVP", in 5th International Conference on Learning Representations, ICLR 2017, 2017.

[99] Kingma, D. P., Dhariwal, P., "Glow: Generative flow with invertible 1x1 convolutions", in Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 2018, str. 10 236–10 245.

[100] Ho, J., Chen, X., Srinivas, A., Duan, Y., Abbeel, P., "Flow++: Improving flow-based generative models with variational dequantization and architecture design", in International Conference on Machine Learning. PMLR, 2019, str. 2722–2730.

[101] Ioffe, S., Szegedy, C., "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, ser. JMLR Workshop and Conference Proceedings, Vol. 37. JMLR.org, 2015, str. 448–456.

[102] Chen, T. Q., Behrmann, J., Duvenaud, D., Jacobsen, J., "Residual flows for invertible generative modeling", in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, str. 9913–9923.

[103] Behrmann, J., Grathwohl, W., Chen, R. T., Duvenaud, D., Jacobsen, J.-H., "Invertible residual networks", in International Conference on Machine Learning. PMLR, 2019, str. 573–582.

[104] Theis, L., van den Oord, A., Bethge, M., "A note on the evaluation of generative models", in International Conference on Learning Representations ICLR, 2016.

[105] Huang, G., Liu, Z., van der Maaten, L., Weinberger, K. Q., "Densely connected convolutional networks", in 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. IEEE Computer Society, 2017, str. 2261–2269.

[106] Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T., "Visualizing the loss landscape of neural nets", in Neural Information Processing Systems, 2018, str. 6391–6401.

[107] Chen, J., Lu, C., Chenli, B., Zhu, J., Tian, T., "Vflow: More expressive generative flows with variational data augmentation", in International Conference on Machine Learning. PMLR, 2020, str. 1660–1669.

[108] Huang, C.-W., Dinh, L., Courville, A., "Augmented normalizing flows: Bridging the gap between generative flows and latent variable models", arXiv preprint arXiv:2002.07101, 2020.

[109] Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Li, Y., Singh, V., "Nyströmformer: A nyström-based algorithm for approximating self-attention", in Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium

on Educational Advances in Artificial Intelligence, EAAI 2021, 2021. AAAI Press, 2021, str. 14 138–14 148.

[110] He, K., Zhang, X., Ren, S., Sun, J., "Deep residual learning for image recognition", in 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, 2016, str. 770–778.

[111] Salimans, T., Goodfellow, I. J., Zaremba, W., Cheung, V., Radford, A., Chen, X., "Improved techniques for training gans", in Neural Information Processing Systems 2016, 2016, str. 2226–2234.

[112] Song, Y., Ermon, S., "Generative modeling by estimating gradients of the data distribution", in Neural Information Processing Systems 2019, NeurIPS 2019, 2019, str. 11 895–11 907.

[113] Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J., "Unrolled generative adversarial networks", in 5th International Conference on Learning Representations, 2017.

[114] Krizhevsky, A., "Learning multiple layers of features from tiny images", University of Toronto, 05 2012.

[115] Liu, Z., Luo, P., Wang, X., Tang, X., "Deep learning face attributes in the wild", in Proceedings of International Conference on Computer Vision (ICCV), December 2015.

[116] Lin, T., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., "Microsoft COCO: common objects in context", in European Conference on Computer Vision, 2014.

[117] Caesar, H., Uijlings, J., Ferrari, V., "Coco-stuff: Thing and stuff classes in context", in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, 2018.

[118] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A., "The pascal visual object classes (voc) challenge", International Journal of Computer Vision, Vol. 88, No. 2, 2009.

[119] Hariharan, B., Arbelaez, P., Bourdev, L. D., Maji, S., Malik, J., "Semantic contours from inverse detectors", in IEEE International Conference on Computer Vision, ICCV, 2011.

[120] Neuhold, G., Ollmann, T., Bulò, S. R., Kontschieder, P., "The mapillary vistas dataset for semantic understanding of street scenes", in IEEE International Conference on Computer Vision, 2017.

[121] Pinggera, P., Ramos, S., Gehrig, S., Franke, U., Rother, C., Mester, R., "Lost and found: detecting small road hazards for self-driving vehicles", in International Conference on Intelligent Robots and Systems, IROS, 2016.

[122] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., "Going deeper with convolutions", in Computer Vision and Pattern Recognition (CVPR), 2015.

[123] Kreso, I., Krapac, J., Segvic, S., "Efficient ladder-style densenets for semantic segmentation of large images", IEEE Trans. Intell. Transp. Syst., Vol. 22, 2021.

[124] Zhu, Y., Sapra, K., Reda, F. A., Shih, K. J., Newsam, S. D., Tao, A., Catanzaro, B., "Improving semantic segmentation via video propagation and label relaxation", in IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2019.

[125] Yu, J. J., Derpanis, K. G., Brubaker, M. A., "Wavelet flow: Fast training of high resolution normalizing flows", in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[126] Perugachi-Diaz, Y., Tomczak, J. M., Bhulai, S., "Invertible densenets with concatenated lipswish", arXiv preprint arXiv:2102.02694, 2021.

[127] Lis, K., Honari, S., Fua, P., Salzmann, M., "Detecting road obstacles by erasing them", CoRR, Vol. abs/2012.13633, 2020.

[128] Lee, K., Lee, K., Lee, H., Shin, J., "A simple unified framework for detecting out-of-distribution samples and adversarial attacks", in Neural Information Processing Systems, NeurIPS, 2018.

[129] Steinhardt, J., Liang, P., "Unsupervised risk estimation using only conditional independence structure", in Neural Information Processing Systems 2016, 2016, str. 3657–3665.

[130] Oza, P., Patel, V. M., "C2ae: Class conditioned auto-encoder for open-set recognition", in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[131] Franchi, G., Bursuc, A., Aldea, E., Dubuisson, S., Bloch, I., "TRADI: tracking deep neural network weight distributions", in 16th European Conference on Computer Vision, ECCV, 2020.

[132] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B., "High-resolution image synthesis with latent diffusion models", in Computer Vision and Pattern Recognition, CVPR, 2022.

[133] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A., "Towards deep learning models resistant to adversarial attacks", in International Conference on Learning Representations, ICLR, 2018.

[134] Gregor, K., Besse, F., Rezende, D. J., Danihelka, I., Wierstra, D., "Towards conceptual compression", in Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, 2016, str. 3549–3557.

[135] Vahdat, A., Macready, W. G., Bian, Z., Khoshaman, A., Andriyash, E., "DVAE++: discrete variational autoencoders with overlapping transformations", in Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, ser. Proceedings of Machine Learning Research, Vol. 80. PMLR, 2018, str. 5042–5051.

[136] Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., Welling, M., "Improving variational inference with inverse autoregressive flow", arXiv preprint arXiv:1606.04934, 2016.

[137] Maaløe, L., Fraccaro, M., Liévin, V., Winther, O., "BIVA: A very deep hierarchy of latent variables for generative modeling", in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, str. 6548–6558.

[138] Sinha, S., Dieng, A. B., "Consistency regularization for variational auto-encoders", CoRR, Vol. abs/2105.14859, 2021.

[139] Ho, J., Jain, A., Abbeel, P., "Denoising diffusion probabilistic models", in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[140] Kim, D., Shin, S., Song, K., Kang, W., Moon, I., "Score matching model for unbounded data score", CoRR, Vol. abs/2106.05527, 2021.

[141] Nichol, A., Dhariwal, P., "Improved denoising diffusion probabilistic models", CoRR, Vol. abs/2102.09672, 2021, dostupno na: https://arxiv.org/abs/2102.09672

[142] Kingma, D. P., Salimans, T., Poole, B., Ho, J., "Variational diffusion models", CoRR, Vol. abs/2107.00630, 2021.

[143] Van Oord, A., Kalchbrenner, N., Kavukcuoglu, K., "Pixel recurrent neural networks", in International Conference on Machine Learning. PMLR, 2016.

[144] Salimans, T., Karpathy, A., Chen, X., Kingma, D. P., "Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications", in 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.

[145] Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., Tran, D., "Image transformer", in Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, ser. Proceedings of Machine Learning Research, Vol. 80. PMLR, 2018, str. 4052–4061.

[146] Chen, X., Mishra, N., Rohaninejad, M., Abbeel, P., "Pixelsnail: An improved autoregressive generative model", in Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, ser. Proceedings of Machine Learning Research, Vol. 80. PMLR, 2018, str. 863–871.

[147] Menick, J., Kalchbrenner, N., "Generating high fidelity images with subscale pixel networks and multidimensional upscaling", in 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, 2019.

[148] Roy, A., Saffar, M., Vaswani, A., Grangier, D., "Efficient content-based sparse attention with routing transformers", Trans. Assoc. Comput. Linguistics, Vol. 9, 2021, str. 53–68.

[149] Bhattacharyya, A., Mahajan, S., Fritz, M., Schiele, B., Roth, S., "Normalizing flows with multi-scale autoregressive priors", in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. Computer Vision Foundation / IEEE, 2020, str. 8412–8421.

[150] Ma, X., Kong, X., Zhang, S., Hovy, E. H., "Macow: Masked convolutional generative flow", in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, str. 5891–5900.

[151] Nielsen, D., Jaini, P., Hoogeboom, E., Winther, O., Welling, M., "Survae flows: Surjections to bridge the gap between vaes and flows", in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[152] Vahdat, A., Kautz, J., “NVAE: A deep hierarchical variational autoencoder”, in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[153] Sadeghi, H., Andriyash, E., Vinci, W., Buffoni, L., Amin, M. H., “Pixelvae++: Improved pixelvae with discrete prior”, CoRR, Vol. abs/1908.09948, 2019.

[154] Razavi, A., van den Oord, A., Poole, B., Vinyals, O., “Preventing posterior collapse with delta-vaes”, in 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, 2019.

[155] Ostrovski, G., Dabney, W., Munos, R., “Autoregressive quantile networks for generative modeling”, in Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, ser. Proceedings of Machine Learning Research, Vol. 80. PMLR, 2018, str. 3933–3942.

[156] Radford, A., Metz, L., Chintala, S., “Unsupervised representation learning with deep convolutional generative adversarial networks”, in 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016.

[157] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A. C., “Improved training of wasserstein gans”, in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, str. 5767–5777.

[158] Zhao, S., Liu, Z., Lin, J., Zhu, J., Han, S., “Differentiable augmentation for data-efficient GAN training”, in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[159] Xiao, Z., Kreis, K., Kautz, J., Vahdat, A., “VAEBM: A symbiosis between variational autoencoders and energy-based models”, CoRR, Vol. abs/2010.00654, 2020.

# Biography

Matej Grcić was born on June 30, 1996, in Šibenik. He completed his primary and secondary education in Drniš. He obtained his undergraduate and graduate degrees in Computer Science from University of Zagreb Faculty of Electrical Engineering and Computing. During his graduate studies in the academic year 2019/20, he was awarded the Rector's Award of the University of Zagreb in the category of individual scientific and artistic work. In 2020, he enrolled in the Doctoral program at the University of Zagreb. As part of his doctoral studies, under the mentorship of Prof. Siniša Šegvić, he researches in the field of machine learning with a special focus on computer vision and publishes papers at prestigious conferences (NeurIPS, ECCV, ICML) and in a journal (IEEE T-PAMI). In 2022, he won the ACDC competition held within the workshops at the CVPR conference. In the academic year 2023/24, he was awarded the prestigious Swiss Government Excellence Scholarship for Foreign Scholars and went on a one-year scientific research stay at the École Polytechnique Fédérale de Lausanne (EPFL). During his doctoral studies, he also undertook short scientific research stays at the University of Wuppertal, Germany, and the Technical University of Prague, Czech Republic. He is a reviewer for top conferences and scientific journals and was part of the program committee for the workshop "VAND 2.0: Visual Anomaly and Novelty Detection - 2nd Edition" held as part of the CVPR conference.

# List of publications

## Journal publications

1. **Grcić, M., Šegvić, S., "Hybrid Open-set Segmentation with Synthetic Negative Data". IEEE Transactions on Pattern Analysis and Machine Intelligence, April 2024**
2. Grcić, M., Bevandić, P., Kalafatić. Z., Šegvić, S., "Dense Out-of-Distribution Detection by Robust Learning on Synthetic Negative Data". Sensors, Vol. 24, January 2024, pp. 1248.

## Conference publications

1. **Grcić, M., Bevandić, P., Šegvić, S., "DenseHybrid: Hybrid Anomaly Detection for Dense Open-Set Recognition". 17th European Conference on Computer Vision, 2022, pp. 500 - 517.**
2. Grcić, M., Bevandić, P., Šegvić, S., "Dense open-set recognition with synthetic outliers generated by Real NVP". International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2021
3. Grcić, M., Grubišić, I., Šegvić, S., "Densely connected normalizing flows". Neural Information Processing Systems, 2021, pp. 23968 - 23982
4. Grcić, M., Šarić, J., Šegvić, S., "On Advantages of Mask-level Recognition for Outlier-aware Segmentation". IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2023, pp. 2937 - 2947
5. Grcić, M., Gadetsky, A., Brbić, M., "Fine-grained Classes and How to Find Them". International Conference on Machine Learning, 2024
6. Delić, A., Grcić, M., Šegvić, S.: "Outlier detection by ensembling uncertainty with negative objectness". The 35th British Machine Vision Conference, 2024

# Životopis

Matej Grcić rođen je 30.06.1996. godine u Šibeniku. Osnovnoškolsko i srednjoškolsko obrazovanje je završio u Drnišu. Preddiplomski i diplomski studij Računarstva završio je na Fakultetu Elektrotehnike i Računarstva Sveučilišta u Zagrebu. Tijekom diplomskog studija u akademskoj godini 2019./20. nagrađen je Rektorovom nagradom Sveučilišta u Zagrebu u kategoriji individualni znanstveni i umjetnički rad. 2020. godine upisuje Doktorski studij Sveučilišta u Zagrebu. U okviru doktorskog studija, pod mentorstvom prof. Siniše Šegvića, istražuje u području strojnog učenja s posebnim fokusom na računalni vid te objavljuje radove na prestižnim konferencijama (NeuriPS, ECCV, ICML) i časopisu (IEEE T-PAMI). 2022. godine osvaja ACDC natjecanje održano u okviru radionica na konferenciji CVPR. U akademskoj godini 2023./24. je u međunarodnoj konkurenciji nagrađen sa prestižnom stipendijom Švicarske Konfederacije za strane studente te odlazi na jednogodišnji znanstveno-istraživački boravak na Ecole Polytechnique Fédérale de Lausanne (EPFL). Tijekom doktorkog studija također odlazi na kraće znanstveno-istraživačke boravke na Sveučilište u Wuppertalu, Njemačka te Tehničko Sveučilište u Pragu, Češka. Recenzent je na vrhunskim konferencijama i znanstvenim časopisima te je bio dio programskog odbora radionice "VAND 2.0: Visual Anomaly and Novelty Detection - 2nd Edition" održane u sklopu konferencije CVPR.