Applications of Generative Approaches for Artificial Intelligence

crafting images and sentences out of thin air

Siniša Šegvić
UniZg-FER D307

Agenda

Part 1: recent advances in generative recognition

- Algorithms for generating complex data
- Applications of generative models

Part 2: recent work in my research group

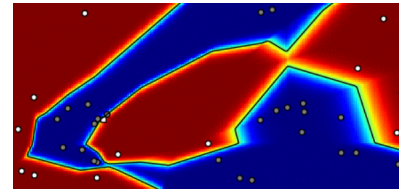
- Overview of anomaly detection
- Generative recognition for dense anomaly detection

[grcic22eccv]

Intro: recognition

Discriminative recognition recovers $P(Y = y|X = x)$

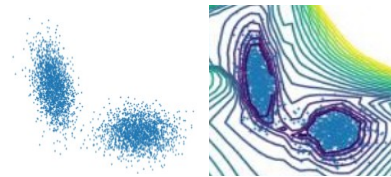
- default flavour of machine learning
- established technology, exciting applications



[unizg-fer-dl1]

Generative recognition recovers $p(X = x, Y = y)$

- or, equally interesting, $p(X = x|Y = y)$ or $p(X = x)$
- called *generative* since sampling from them generates **synthetic data** in the input space
- somewhat eclipsed by the success of discriminative approaches
- rapidly developing, exciting applications

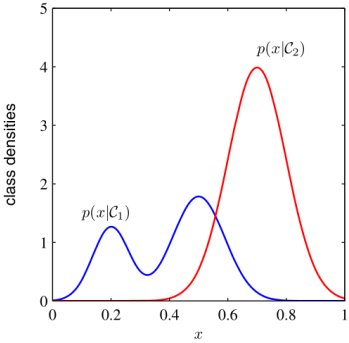


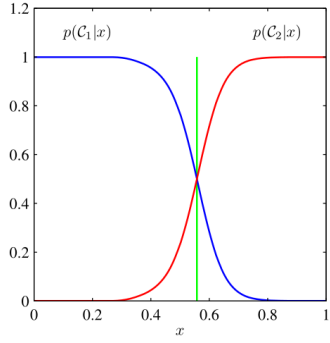
[delic21sem]

Intro: generative vs discriminative

Both approaches produce **probabilistic** output in each input datum, however:

- discriminative models produce distributions over **targets**
- generative models produce distributions over **inputs**
  - how to normalize the distribution, i.e. ensure that it sums (integrates) to 1?
- generative recognition is tough!

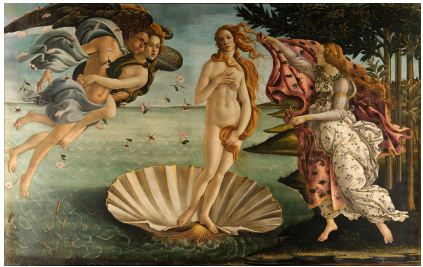




[bishop06book]

Intro: generative vs discriminative (2)

Intuitively, it is easier to tell Boticelli from Picasso than to actually do the painting:



[public domain]

It is easier to distinguish composers than to develop musical ideas.

In words of the infamous food critic Anton Ego:

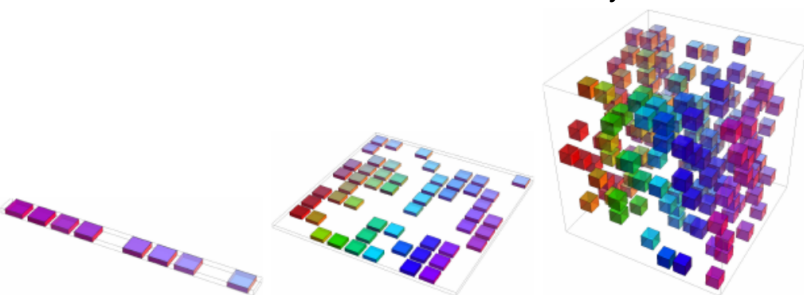
> *... in the grand scheme of things, the average piece of (bad food) is probably more meaningful than our criticism designating it so.*

Intro: generative vs discriminative (3)

Things get especially tough when the data is complex:

- language: 15-20 words per sentence, 170000 total words (English)
- vision: at least $3 \times 64 \times 64$ components, 256 values per component
- generative recognition has to consider integrals over thousands of dimensions in order to normalize $p(\mathbf{x})$

Close encounters with the curse of dimensionality!



Intro : tasks

Generative approaches aim at density of the training data

Three main tasks of a generative model:

- generate synthetic data points by sampling $p(\mathbf{x})$
  - trade off: quality vs coverage
  - useful for **content creation** and **improving** discriminative models
- transform data-points to a factorized latent representation
  - useful for **content editing**
- evaluate density $p(\mathbf{x})$
  - only generative models with explicit density can do that
  - useful for **anomaly detection**, **compression**

Intro : approaches

Generative algorithms come in many flavours and many ways to appreciate them:

Algorithm	latent	bias	sampling	density
Energy	unable	coverage	slow	unnormalized
VAE	easy	coverage	fast	ELBO
Autoreg.	unable	coverage	slow	exact
GAN	unable	quality	fast	unable
NFlow	easy	coverage	fast	exact
Diffusion	easy	(coverage)	slow	ELBO
Score	unable	coverage	slow	(unable)

Intro: density estimation

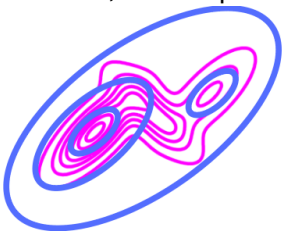
Comparison of generative algorithms with respect to efficient density estimation:

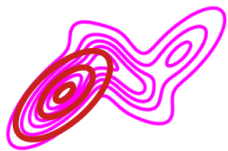
algorithm / density formulation	tractability	efficiency
$p_{EBM}(\mathbf{x}) = e^{-E(\mathbf{x})} / \int_{\mathbf{x}} e^{-E(\mathbf{x})}$	**tractable inference**	**fast**
$p_{VAE}(\mathbf{x}) = \int_{\mathbf{z}} p(\mathbf{x}\|\mathbf{z}) p(\mathbf{z}) d\mathbf{z}$	intractable (ELBO)	fast
$p_{\mathrm{ar}}(\mathbf{x}) = p(x_1) \prod_{i=2}^{HW} p(x_i\|\mathbf{x}_{<i})$	tractable	slow
$p_{GAN}(\mathbf{x}) =?$ (implicit density)	unavailable	
$p_{flow}(\mathbf{x}) = p_z(\mathbf{f}(\mathbf{x})) \cdot \lvert \det(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}) \rvert$	**tractable**	**fast**
$p_{diff}(\mathbf{x}) = \int p(\mathbf{x}\|\mathbf{x}^{(1)})$ $\prod_{t=1}^{T-1} p(\mathbf{x}^{(t)}\|\mathbf{x}^{(t+1)}) \pi(\mathbf{x}^T) d\mathbf{x}^{(1\ldots T)}$	intractable (ELBO)	slow

Applications : content creation

Most generative algorithms are able to generate data.

However, some optimize for coverage while others optimize for quality [lucas19neurips].





Normalizing flows - high coverage (left); GANs - high quality (right).

[grcic21neurips]

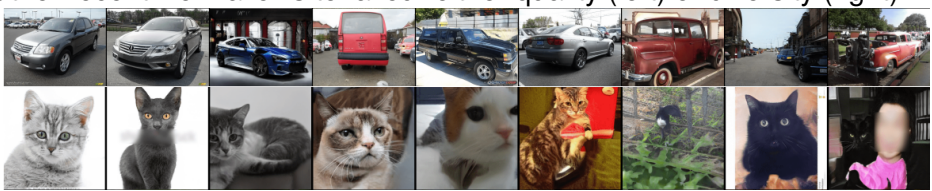
[sehwag22cvpr]

Applications : boosting diversity

Recent work leverages computational power to boost diversity without sacrificing quality:

[sehwag22cvpr]

Other recent work allows to favour either quality (left) or diversity (right):



[humayun22cvpr]

Applications : conditional generation

For practical puprposes, we are most interested in conditional generation

A popular recent approach connects language embeddings with generative vision.

This is what I got by feeding "a photo of a white cat on a unicycle" to DALL-e:

[DALL-E]

Applications : conditional generation (2)

It also works the other way round (from images to text):

GT: A young boy in the park throwing a frisbee.

L-Verse: A young boy throwing a green frisbee in a lush green park.

GT: A laptop and a cell phone on a table.

L-Verse: A collection of electronic devices and cords sitting on top of a table.

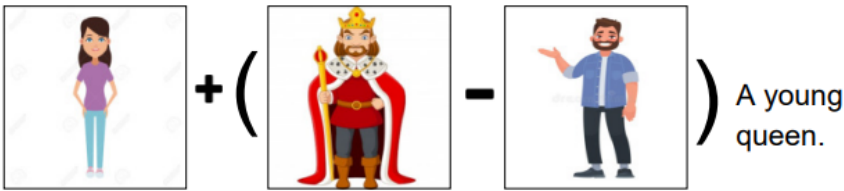
GT: A small bathroom is shown from a door.

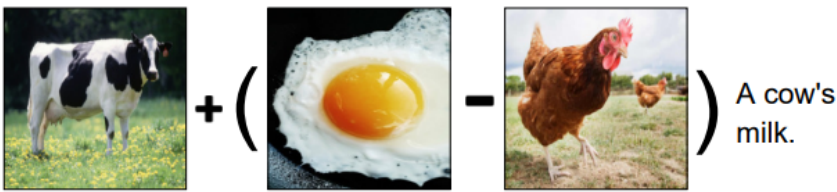
L-Verse: A bathroom with a shower curtain over the bathtub next to a toilet.

[kim22cvpr]

Applications : conditional generation (3)

One can also perform arithmetic operations on visual semantics and display results in text:

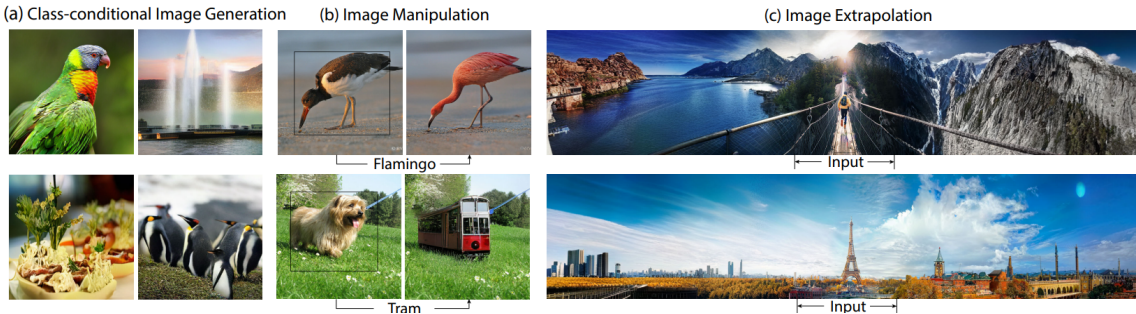




[tewel22cvpr]

Applications : content editing

Conditional generation, visual replace, extrapolation by editing the VQ VAE latent:



[chang22cvpr]

Applications : content editing (2)

Another instance of visual replace:

[couairon22cvpr]

Applications : content editing (3)

Super resolution by leveraging Style GAN latent:

(a) Input LR (b) GPEN [30] (c) GLEAN [3] (d) IRN [28] (e) Ours (f) GT

[zhong22cvpr]

Applications : content editing (4)

Colorization and other inverse problems (inpainting, medical image reconstruction):

[song22blog]

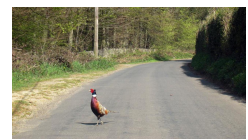
Applications: anomaly detection

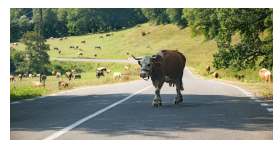
Detect images (or pixels) that are unrelated to the training data

Our benchmark: Segment Me If You Can [chan21neurips]

The task is to detect pixels that do not belong to any of the 19 road-driving classes:



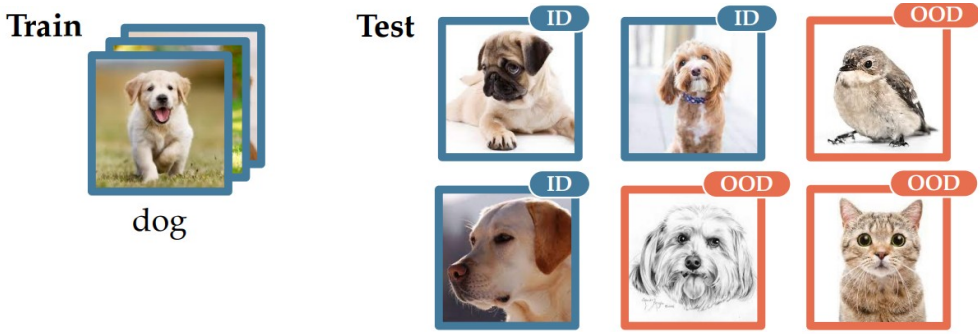



[https://segmentmeifyoucan.com/]

Anomaly detection: about anomalies

Definition: an observation which arouses suspicions of being unrelated to the process that generates training data [hawkins80book].



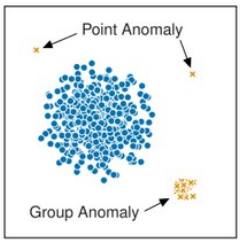
[yang21arxiv]

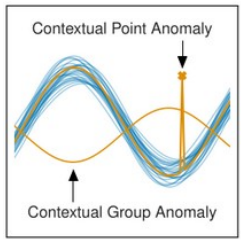
Anomalous data points are related (or also known as) outliers, out-of-distribution samples or novelties [ruff21pieee].

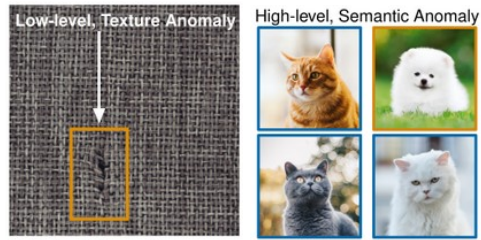
Anomaly detection: about anomalies (2)

There are several kinds of anomalies:

- pointwise vs groupwise
- contextual pointwise vs contextual groupwise
- low level (texture) vs high level (semantics)





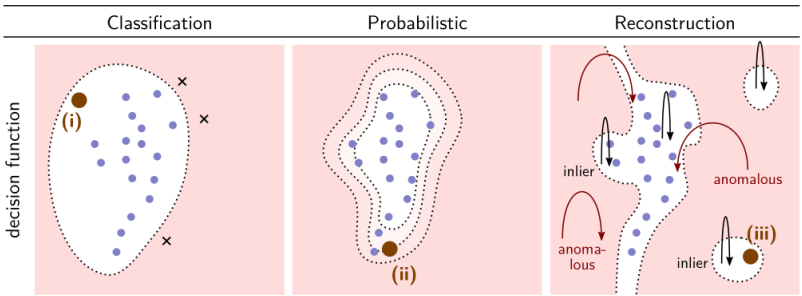


[ruff21pieee]

Anomaly detection: overview

Three main approaches to express anomaly score s of the sample $\mathbf{x}$:

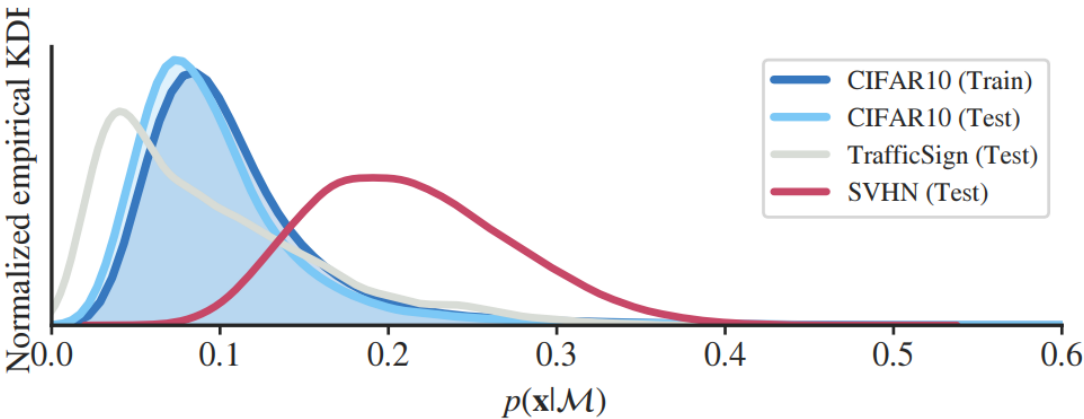
- arbitrary scalar $s_{\text{rec}} = f(\mathbf{x})$ ("classification")
  - related to dataset posterior $P(\mathcal{D}_{\text{in}}|\mathbf{x}) = \sigma(s_{\text{cls}}))$
- probability density function $s_{\text{pdf}} = p(\mathbf{x})$ ("probabilistic")
- reconstruction error $s_{\text{rec}} = \|\mathbf{x} - f_{\text{dec}}(f_{\text{enc}}(\mathbf{x})\|$ ("reconstruction")
  - related to unnormalized density $P(\mathcal{D}_{\text{in}}|\mathbf{x}) \sim exp(-s^2_{\text{rec}}))$



[ruff21pieee]

Anomaly detection: reality check

Direct application of estimated density (either flows or pixel-cnn) to outlier detection fails miserably:

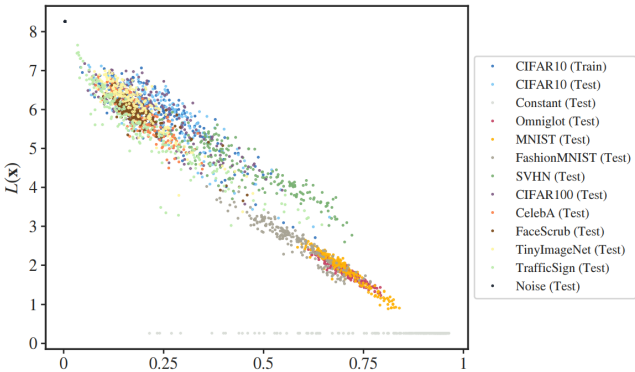


[serra20iclr]

Anomaly detection: role of complexity

Recovered densities wildly depend on image complexity:

- consider images with lower compressed lengths $L(\mathrm{x})$
- e.g. MNIST, poliglot, constant (the simple ones)
- these images give rise to higher densities in spite of being outliers

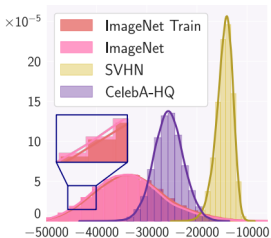


[serra20iclr]

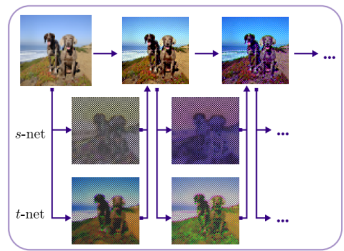
Anomaly detection: inappropriate bias

Possible explanation: maximum likelihood training is unable to recover semantic anomalies since generative models know nothing about semantics.

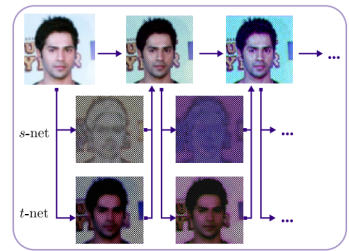
- visualization of internal activations suggest a similar reaction for inliers and outliers
- however, they train only with generative loss $L = -\log p(\mathbf{x})$
- chances improve when sharing features with a discriminative task [zhang00eccv]



(a) Log-likelihoods



(b) ImageNet input, in-distribution



(c) CelebA input, OOD

[kirichenko20neurips]

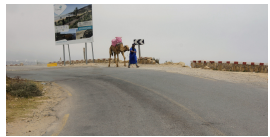
Dense anomalies: generative aproaches

Detect regions with low pixel-level density:

- apply any generative model to 1x1 feature windows [blum21ijcv]
- find a way to train EBM without sampling - DenseHybrid [grcic22eccv]

Leverage generative modeling in non-density based approaches:

- detect reconstruction errors in the resynthesized image [lis19iccv]
- discriminative training with jointly generated synthetic negatives - NFlowJS [grcic21visapp, grcic21arxiv]



[https://segmentmeifyoucan.com/]

Dense anomalies: pixel-level density

Dense density estimation: recover probability density as if in a sliding window

Desireable properties: efficient inference, equivariance to translation

- VAE - **fast**, not equivariant (vector latent)
- EBM - **fast**, **equivariant** (but intractable training)
- pixel-cnn - slow, not equivariant ("linear" factorization)
- flow - **fast**, not equivariant
- diffusion, score-based - slow, not equivariant
- GAN - fast, can be equivariant (but no explicit density)

Dense anomalies: sliding 1×1 window

Apply per-layer flows to frozen features of a standard semantic segmentation model

- embedding density [blum21ijcv]

The training optimizes normalized log-likelihood of features $z_\ell^{(i)}$ for layer ℓ and batch index i: $\overline{N}(z_\ell^{(i)}) = \log p(z_\ell^{(i)}) - \frac{1}{N} \sum_k \log p(z_\ell^{(k)})$

Strength: combines principled density estimation with feature semantics

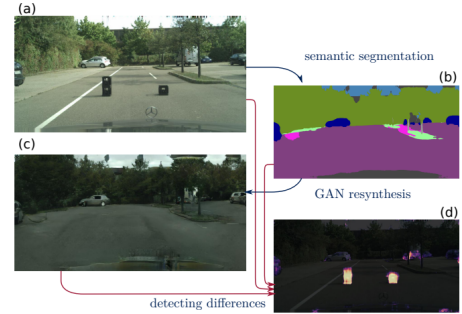
Weakness 1: vulnerability to feature collapse due to frozen features

Weakness 2: does not exploit negative training data

Dense anomalies: image resynthesis

Approach [lis19iccv, vojir21iccv, dibiase21cvpr]:

1. perform standard semantic segmentation
2. resynthesize input by generative image-to-image translation
3. detect anomalous pixels as reconstructions errors



[lis19iccv]

Strength: rather principled, can detect all kinds of anomalies

Weakness 1: suitable only for anomalies on the road

Weakness 2: semantics-to-RGB fails in non-standard road pixels

Weakness 3: rather slow, unsuitable for real-time

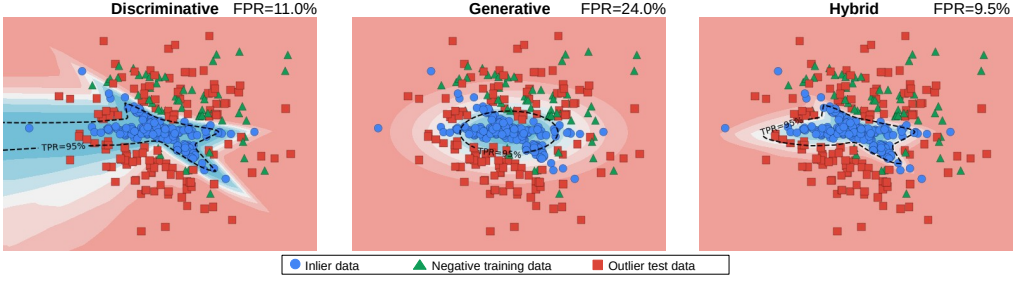
Dense anomalies: equivariant density

Start with any model for discriminative dense prediction

Detect anomalies according to energy-based pixel-level density [grcic22eccv]

Train (or fine-tune) with discriminative and generative loss

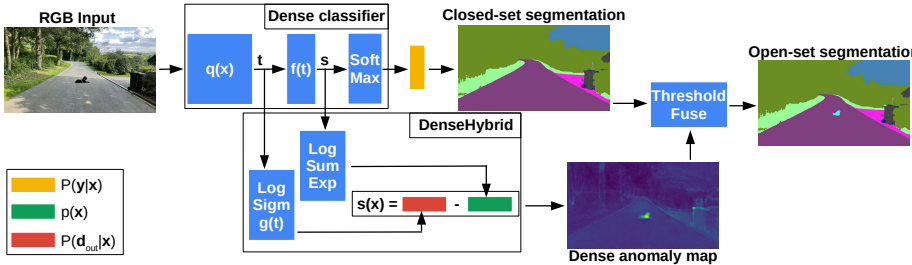
- joint hybrid training provides a chance to avoid feature collapse
- further improve by exploiting noisy negative data from ADE20k [bevandic22ivc]



Dense anomalies: equivariant density (2)

Use an energy based-model for likelihood-based scoring

- avoid intractable normalization constant through training with noisy negative examples
- strength: neglectable computational overhead wrt discriminative baseline
- strength: state of the art performance on standard benchmarks
- weakness: poor quality of generated data
  - less important in our setup, we did not even try



Dense anomalies : synthetic negatives

Training with pasted noisy negative samples produces great outlier detection performance

- hard to evaluate the performance
- some test anomalies may have been seen during training...

We wish to address this issue by replacing real negative samples with synthetic ones

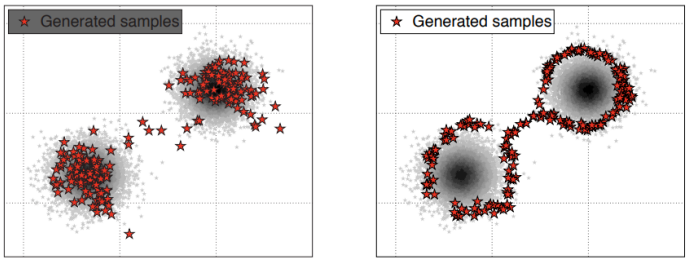
Question: how to co-train a generative model in order to produce negative examples which could teach the discriminative model to better recognize anomalies?

Dense anomalies: synthetic negatives (2)

We pose the following requirements on model parameters θ:

- high data likelihood in inliers $p_\theta(\mathrm{x})$
- high discriminative entropy in generated data $P(y|\mathbf{f}_\theta(\mathrm{z}))$

Such learning generates samples at the border of the training distribution [lee18iclr]



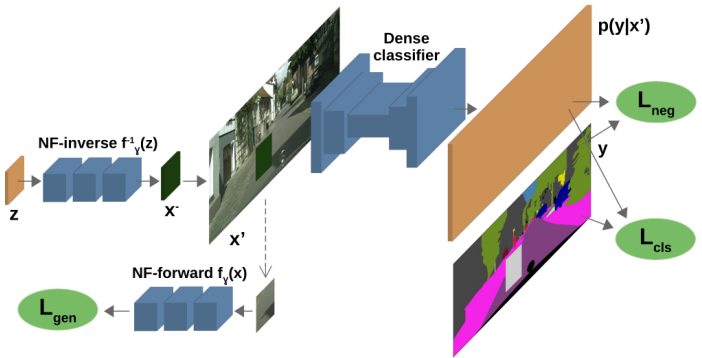
[lee18iclr]

The dicriminative model can be trained to predict high uncertainty in these samples!

Dense anomalies: synthetic negatives (3)

We adapt the joint learning scheme for dense prediction:

- we use a normalizing flow instead of GAN (arbitrary resolution, better coverage)
- we contribute a robust loss that accounts for generative noise
- we propose a suitable optimization procedure for joint learning



[grcic21visapp]

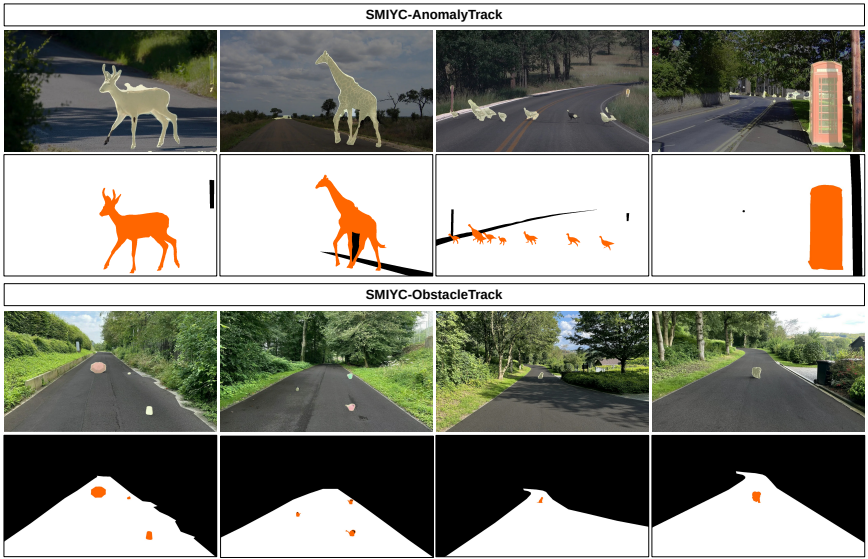
Dense anomalies: experimental evaluation

Anomaly Track

Method	OoD Data	Pixel Level		Component Level		
		AUPR	FPR_{95}	sIoU gt	PPV	mean F1
Maximized Entropy [paper] [code]	✓	85.47%	15.00%	49.21%	39.51%	28.72%
DenseHybrid [paper] [code]	✓	77.96%	9.81%	54.17%	24.13%	31.08%
ObsNet [paper] [code]	✗	75.44%	26.69%	44.22%	52.56%	45.08%
NFlowJS [paper]	✗	56.92%	34.71%	36.94%	18.01%	14.89%
SynBoost [paper] [code]	✓	56.44%	61.86%	34.68%	17.81%	9.99%
Image Resynthesis [paper] [code]	✗	52.28%	25.93%	39.68%	10.95%	12.51%
Embedding Density [paper]	✗	37.52%	70.76%	33.86%	20.54%	7.90%
Void Classifier [paper]	✓	36.61%	63.49%	21.14%	22.13%	6.49%
JSRNet [paper] [code]	✗	33.64%	43.85%	20.20%	29.27%	13.66%
ODIN [paper]	✗	33.06%	71.68%	19.53%	17.88%	5.15%
MC Dropout [paper]	✗	28.87%	69.47%	20.49%	17.26%	4.26%
Maximum Softmax [paper]	✗	27.97%	72.05%	15.48%	15.29%	5.37%
Mahalanobis [paper]	✗	20.04%	86.99%	14.82%	10.22%	2.68%

[https://segmentmeifyoucan.com/]

Dense anomalies: qualitative experiments



[grcic22eccv]

Conclusion

- Generative recognition has experienced a lot of exciting recent progress.
  - We are proud of our systems although they are not intelligent in the strong sense.
- Image is a collection of easily counterfeited pixels
  - verification of integrity possible only in presence of cryptographic signatures
  - important implications for our society
- Semantic anomaly detection can not be properly addressed in absence of semantic supervision.
- Open-set recognition appears easier than four years ago
  - it is not unlikely that soon it will be considered as solved.

Thank you for your attention!

Questions?

This presentation would not have been possible without insightful ideas and hard work of Matej Grcić, Jakob Verbeek, Ivan Krešo, Marin Oršić, Petra Bevandić, Josip Šarić, Ivan Grubišić, Marin Kačan, Iva Sović, Nenad Markuš and Jelena Bratulić.

This research has been supported by Croatian Science Foundation (MULTICLOD, ADEPT), ERDF (DATACROSS, A-UNIT, SAFETRAM), NVidia Academic Hardware Grant Program, Rimac automobili, Microblink, Gideon brothers, Romb technologies, Promet i prostor, Končar, UniZg-FPZ, and VSITE.