Backpropagated Gradient Representations for Anomaly Detection

Gukyeong Kwon* Mohit Prabhushankar Dogancan Temel Ghassan AlRegib

(*: Speaker)

Georgia Institute of Technology

August 2020

Georgia Tech
CREATING THE NEXT

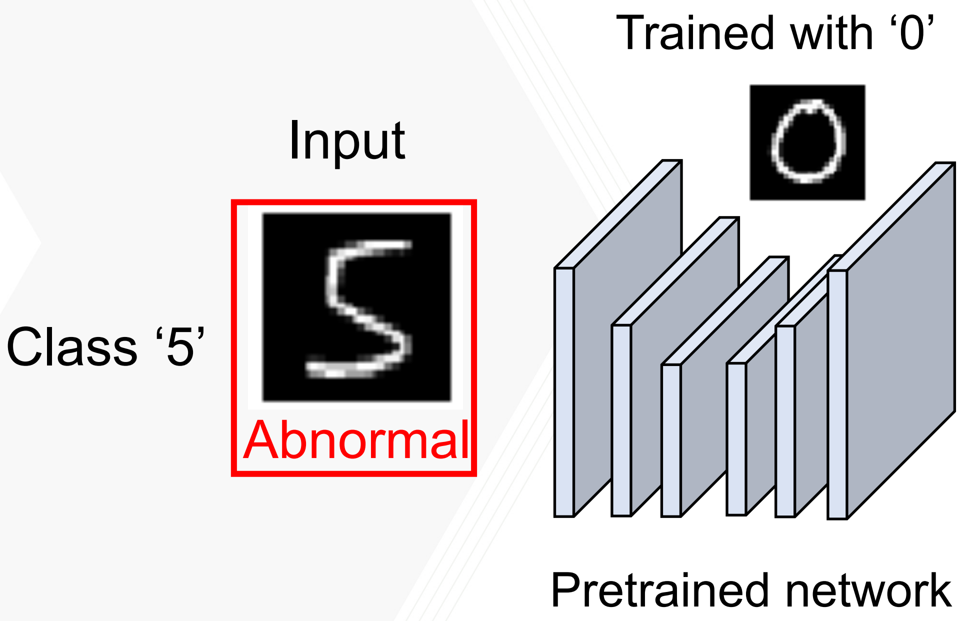
ECCV'20 ONLINE 23-28 AUGUST 2020

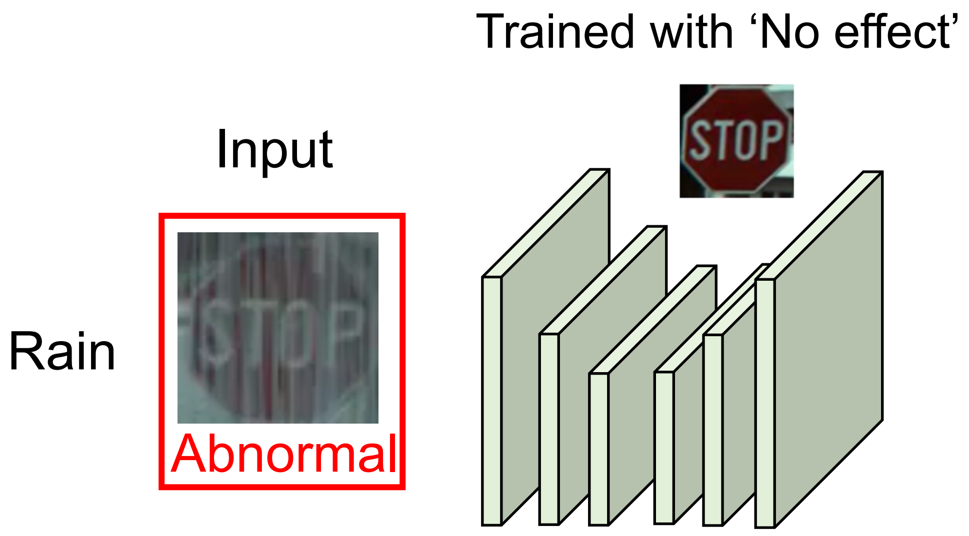
OLIVES @GeorgiaTech

Paper & codes

Overview
Anomaly Detection

Anomaly: Data whose *classes* or *attributes* differs from training data

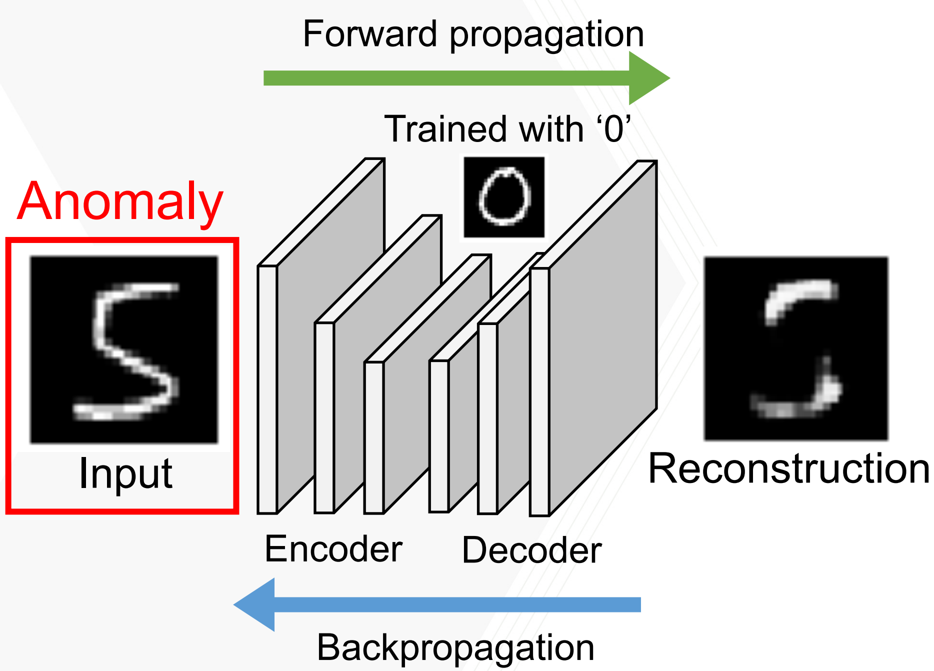




Goal: Detect anomalies to ensure the robustness of machine learning algorithm

Overview

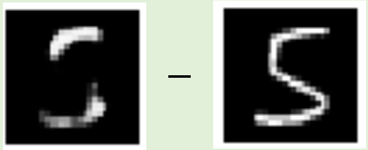
Gradient-based Representation



Existing approaches

Activation-based representation
(Data perspective)

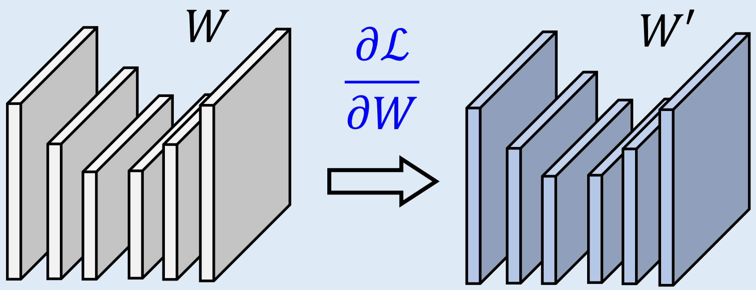
e.g. Reconstruction error ($\mathcal{L}$)



How much of the input does not correspond to the learned information?

Proposed approach

Gradient-based Representation
(**Model** perspective)



How much **model update** is required by the input?

Contributions

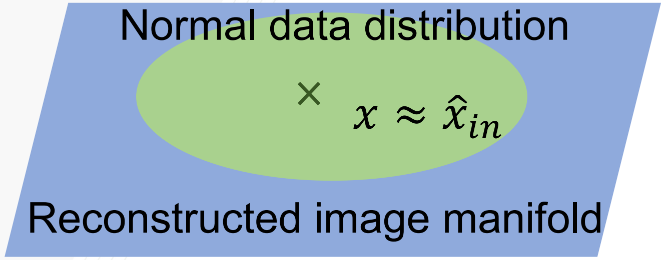
1. We propose utilizing backpropagated gradients as representations to characterize anomalies.

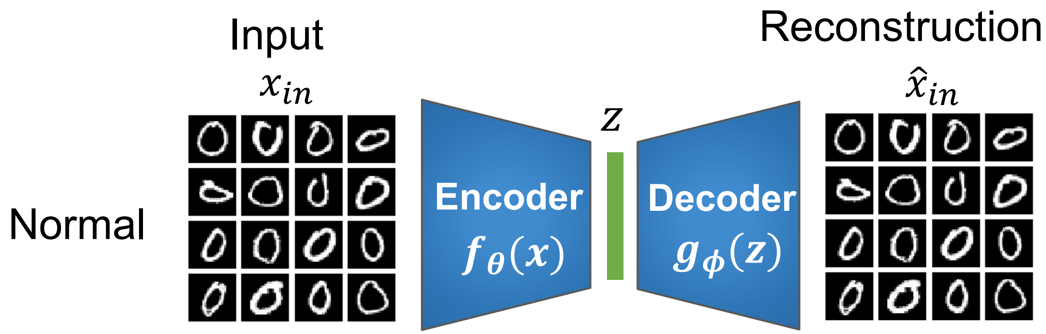
2. We validate the representation capability of gradients for anomaly detection in comparison with activation through comprehensive baseline experiments.

3. We propose an anomaly detection algorithm using gradient-based representations and show that it outperforms state-of-the-art algorithms using activation-based representations.

Geometric Interpretation

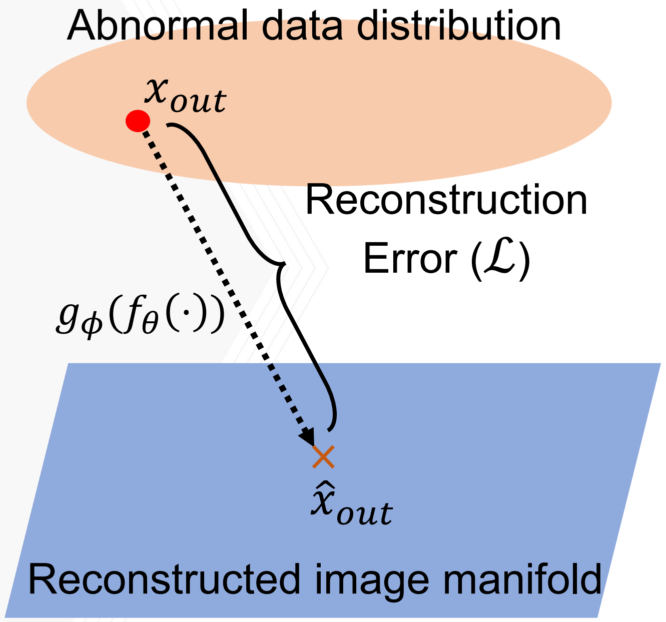
Advantages of Gradient-based Representations

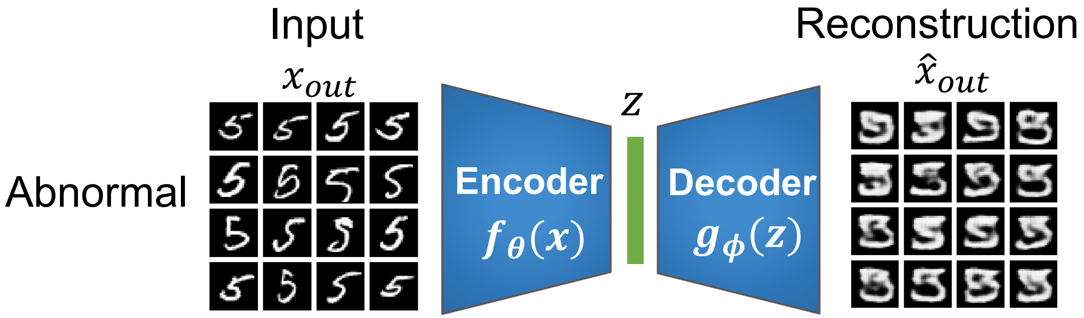




Geometric Interpretation

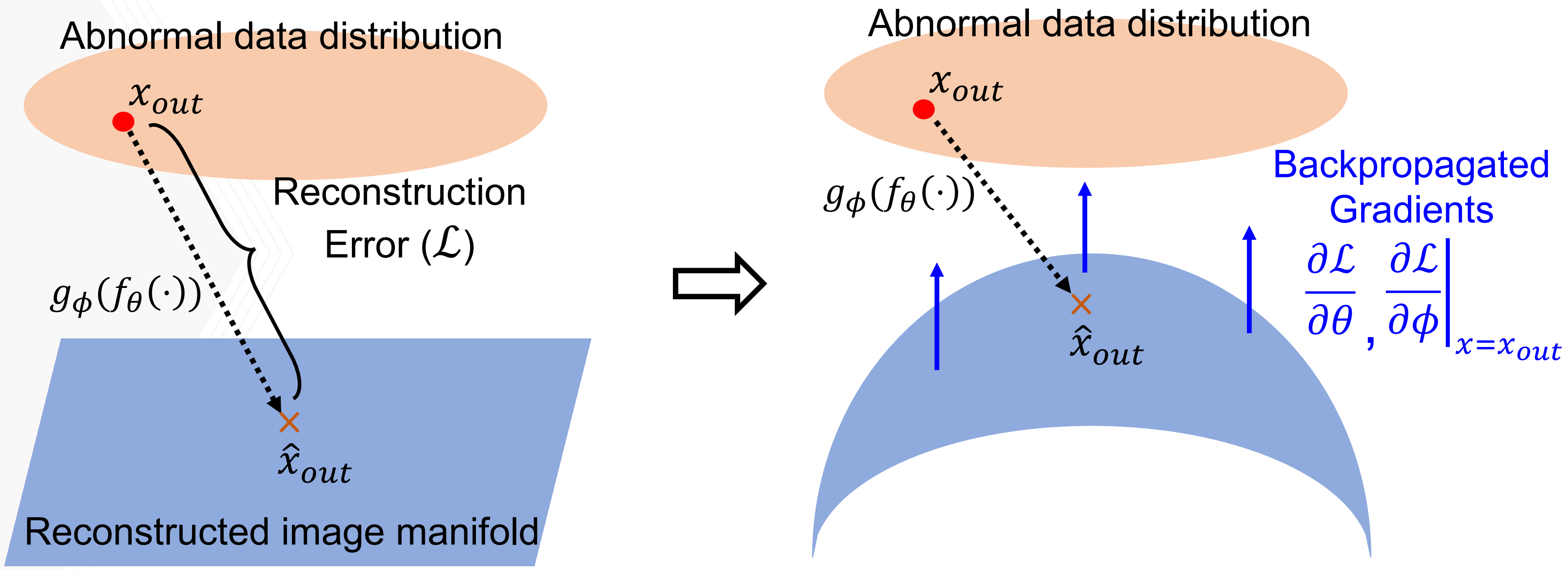
Advantages of Gradient-based Representations





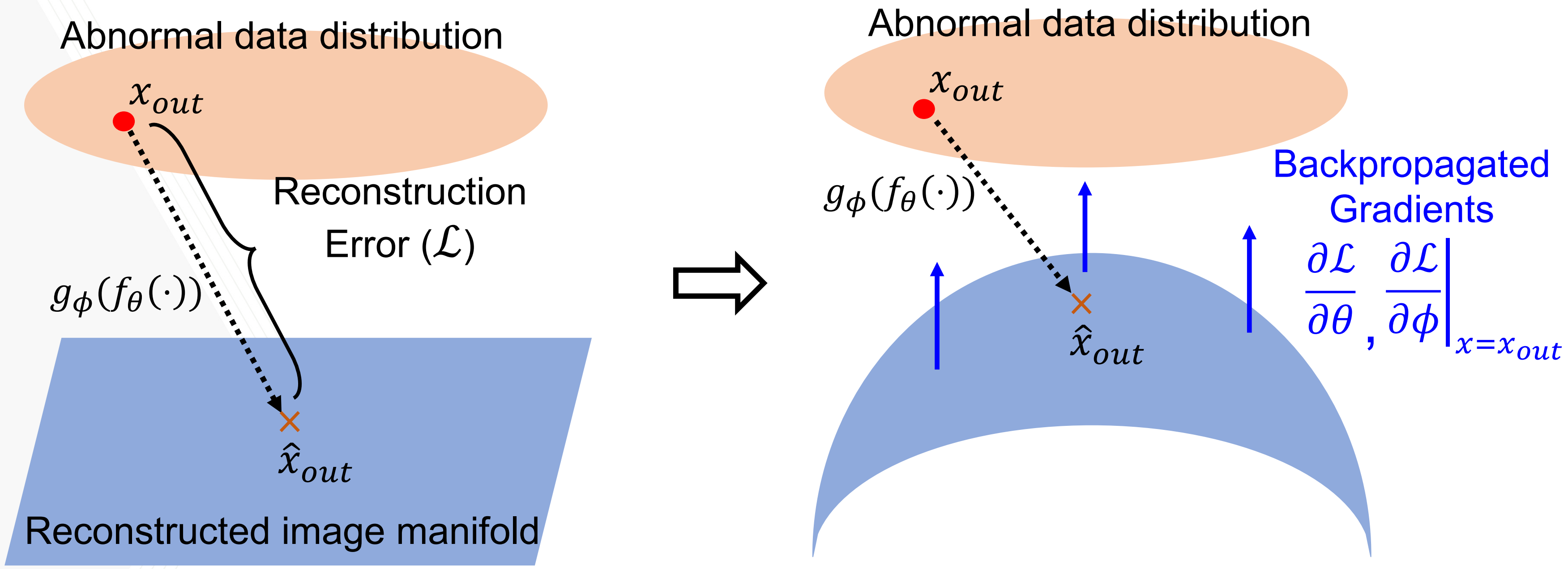
Geometric Interpretation

Advantages of Gradient-based Representations



Geometric Interpretation

Advantages of Gradient-based Representations

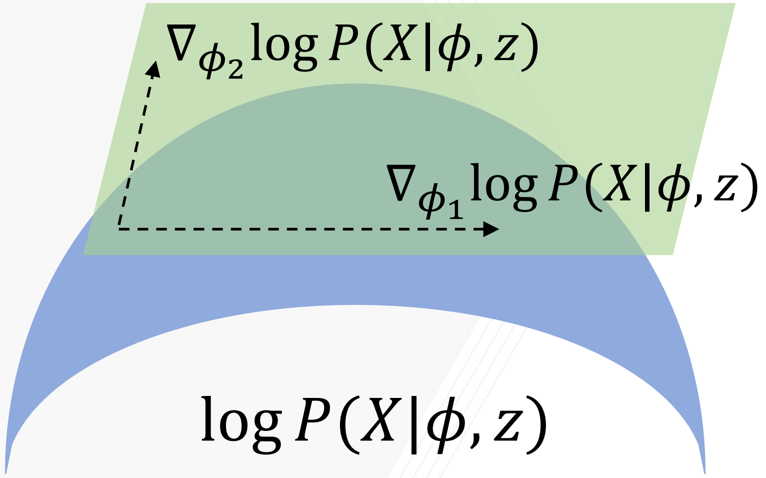


1) Provide directional information to characterize anomalies

2) Gradients from different layers capture abnormality at different levels of data abstraction

Theoretical Interpretation

Fisher Kernel



ϕ : Decoder weight

z : Latent variable

Measure difference between two data points (X_i, X_j)

Fisher kernel $$K_{FK}(X_i, X_j) = {U_{\phi}^{X_i}}^T F^{-1} U_{\phi}^{X_j}$$

Fisher score $$\mathrm{U}_{\phi}^{X} = \nabla_{\phi} \log P(X|\phi, z)$$

Fisher information matrix $$F = \mathrm{E}_X[U_{\phi}^{X} {U_{\phi}^{X}}^T]$$

Theoretical Interpretation
Fisher Kernel

Distance between normal data

$$K_{FK}^{in}(X_{tr}, X_{te,in}) = {U_{\phi}^{X_{tr}}}^{T} F^{-1} \boxed{U_{\phi}^{X_{te,in}}}$$

X_{tr}: Training data (normal)

$X_{te,in}$: Test normal data

Distance between normal and abnormal data

$$K_{FK}^{out}(X_{tr}, X_{te,out}) = {U_{\phi}^{X_{tr}}}^{T} F^{-1} \boxed{U_{\phi}^{X_{te,out}}}$$

X_{tr}: Training data (normal)

$X_{te,in}$: Test abnormal data

For anomaly detection,

$$K_{FK}^{out}(X_{tr}, Y_{te,out}) \gg K_{FK}^{in}(X_{tr}, X_{te,in})$$

When the autoencoder is trained to minimize negative loglikelihood loss,

$$\frac{\partial \mathcal{L}}{\partial \phi} \rightarrow \mathrm{U}_{\phi}^{X} = \nabla_{\phi} \log P(X|\phi, z)$$

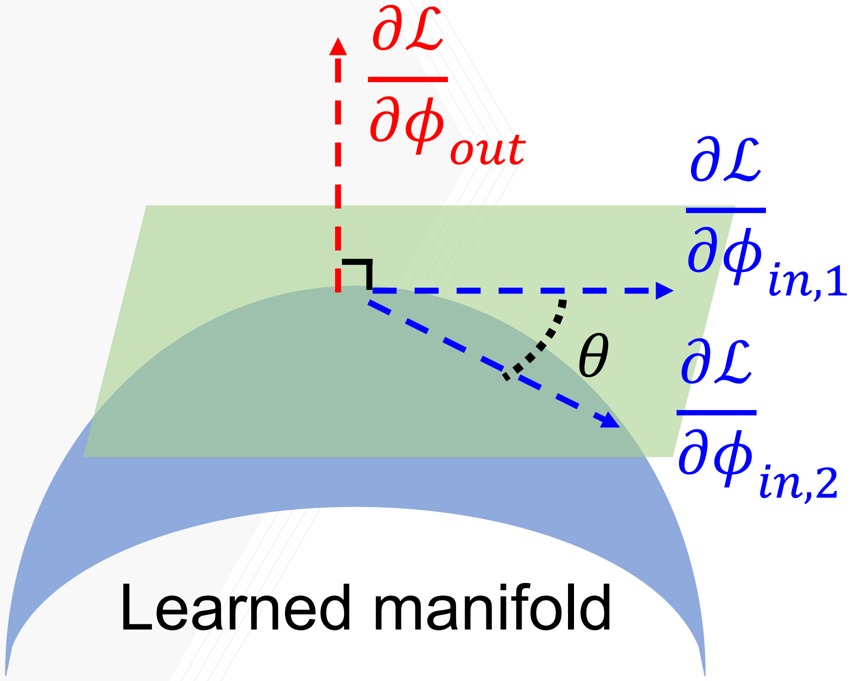
→ Backpropagated gradients are descriptive representations for anomalies

OLIVES @GeorgiaTech www.ghassanalregib.com

Georgia Tech CREATING THE NEXT

GradCon: Gradient Constraint

Constrain gradient-based representations during training to obtain clear separation between normal data and abnormal data



ϕ: Weights $\mathcal{L}$: Reconstruction error

At k-th step of training,

$$J = \mathcal{L} - \overbrace{\mathbb{E}_i\left[\text{cosSIM}\left(\frac{\partial J}{\partial \phi_{i_{avg}}}^{k-1}, \frac{\partial \mathcal{L}}{\partial \phi_i}^{k}\right)\right]}^{\text{Gradient loss}}$$

Avg. training gradients until (k-1) th iter.

Gradients at k-th iter.

where $$\frac{\partial J}{\partial \phi_{i_{avg}}}^{k-1} = \sum_{t=1}^{k-1} \frac{\partial J}{\partial \phi_i}^{t}$$

Baseline Experiment
Activation vs. Gradients

Abnormal "class" detection (CIFAR-10)

e.g.

Normal

Abnormal

AUROC Results

Model	Loss	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Average
CAE	Recon	0.682	0.353	0.638	0.587	0.669	**0.613**	0.495	0.498	0.711	0.390	0.564
CAE + Grad	Recon	0.659	0.356	**0.640**	0.555	0.695	0.554	0.549	0.478	0.695	0.357	0.554
	Grad	**0.752**	0.619	0.622	0.580	0.705	0.591	0.683	**0.576**	**0.774**	**0.709**	**0.661**
VAE	Recon	0.553	0.608	0.437	0.546	0.393	0.531	0.489	0.515	0.552	0.631	0.526
	Latent	0.634	0.442	**0.640**	0.497	**0.743**	0.515	**0.745**	0.527	0.674	0.416	0.583
VAE + Grad	Recon	0.556	0.606	0.438	0.548	0.392	0.543	0.496	0.518	0.552	0.631	0.528
	Latent	0.586	0.396	0.618	0.476	0.719	0.474	0.698	0.537	0.586	0.413	0.550
	Grad	0.736	**0.625**	0.591	**0.596**	0.707	0.570	0.740	0.543	0.738	0.629	0.647

Recon: Reconstruction error, Latent: Latent loss, Grad: Gradient loss

1) (CAE vs. CAE + Grad) Effectiveness of the gradient constraint

Baseline Experiment
Activation vs. Gradients

Abnormal "class" detection (CIFAR-10)

e.g.

Normal

Abnormal

AUROC Results

Model	Loss	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Average
CAE	Recon	0.682	0.353	0.638	0.587	0.669	**0.613**	0.495	0.498	0.711	0.390	0.564
CAE + Grad	Recon	0.659	0.356	**0.640**	0.555	0.695	0.554	0.549	0.478	0.695	0.357	0.554
	Grad	**0.752**	0.619	0.622	0.580	0.705	0.591	0.683	**0.576**	**0.774**	**0.709**	**0.661**
VAE	Recon	0.553	0.608	0.437	0.546	0.393	0.531	0.489	0.515	0.552	0.631	0.526
	Latent	0.634	0.442	**0.640**	0.497	**0.743**	0.515	**0.745**	0.527	0.674	0.416	0.583
VAE + Grad	Recon	0.556	0.606	0.438	0.548	0.392	0.543	0.496	0.518	0.552	0.631	0.528
	Latent	0.586	0.396	0.618	0.476	0.719	0.474	0.698	0.537	0.586	0.413	0.550
	Grad	0.736	**0.625**	0.591	**0.596**	0.707	0.570	0.740	0.543	0.738	0.629	0.647

Recon: Reconstruction error, Latent: Latent loss, Grad: Gradient loss

1) (CAE vs. CAE + Grad) Effectiveness of the gradient constraint

2) (CAE vs. VAE) Performance sacrifice from the latent constraint

Baseline Experiment

Activation vs. Gradients

Abnormal “class” detection (CIFAR-10)

e.g.

Normal

Abnormal

AUROC Results

Model	Loss	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Average
CAE	Recon	0.682	0.353	0.638	0.587	0.669	**0.613**	0.495	0.498	0.711	0.390	0.564
CAE + Grad	Recon	0.659	0.356	**0.640**	0.555	0.695	0.554	0.549	0.478	0.695	0.357	0.554
	Grad	**0.752**	0.619	0.622	0.580	0.705	0.591	0.683	**0.576**	**0.774**	**0.709**	**0.661**
VAE	Recon	0.553	0.608	0.437	0.546	0.393	0.531	0.489	0.515	0.552	0.631	0.526
	Latent	0.634	0.442	**0.640**	0.497	**0.743**	0.515	**0.745**	0.527	0.674	0.416	0.583
VAE + Grad	Recon	0.556	0.606	0.438	0.548	0.392	0.543	0.496	0.518	0.552	0.631	0.528
	Latent	0.586	0.396	0.618	0.476	0.719	0.474	0.698	0.537	0.586	0.413	0.550
	Grad	0.736	**0.625**	0.591	**0.596**	0.707	0.570	0.740	0.543	0.738	0.629	0.647

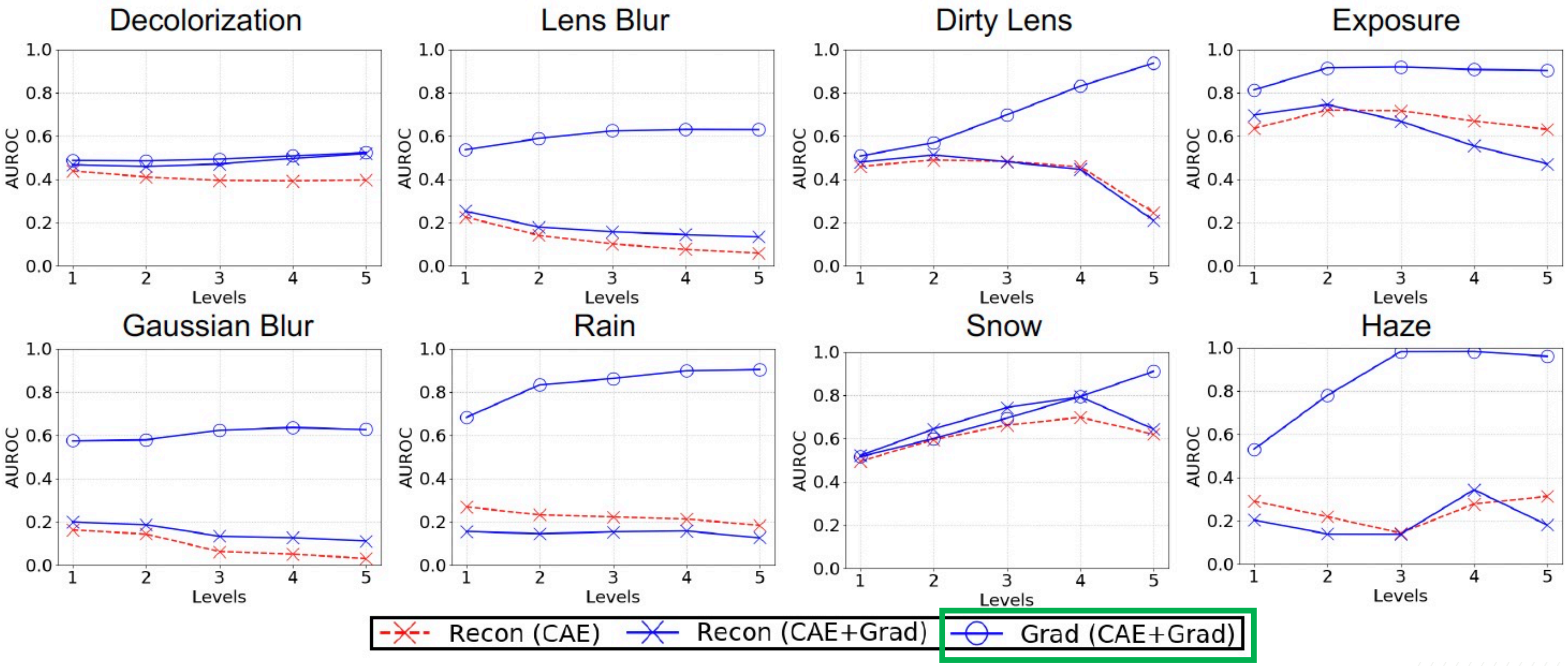
Recon: Reconstruction error, Latent: Latent loss, Grad: Gradient loss

1) (CAE vs. CAE + Grad) Effectiveness of the gradient constraint
2) (CAE vs. VAE) Performance sacrifice from the latent constraint
3) (VAE vs. VAE + Grad) Complementary features from the gradient constraint

Baseline Experiment
Abnormal Condition detection

Abnormal “condition” detection (CURE-TSR)

AUROC Results



Recon: Reconstruction error, Grad: Gradient loss

Comparison with State-of-The-Art Algorithms

CIFAR-10, MNIST, Fashion MNIST

AUROC results in CIFAR-10

	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Average
OCSVM [34]	0.630	0.440	0.649	0.487	0.735	0.500	0.725	0.533	0.649	0.508	0.586
KDE [4]	0.658	0.520	**0.657**	0.497	0.727	0.496	**0.758**	0.564	0.680	0.540	0.610
DAE [9]	0.411	0.478	0.616	0.562	0.728	0.513	0.688	0.497	0.487	0.378	0.536
VAE [12]	0.634	0.442	0.640	0.497	**0.743**	0.515	0.745	0.527	0.674	0.416	0.583
PixelCNN [20]	**0.788**	0.428	0.617	0.574	0.511	0.571	0.422	0.454	0.715	0.426	0.551
LSA [1]	0.735	0.580	**0.690**	0.542	**0.761**	0.546	**0.751**	0.535	0.717	0.548	0.641
AnoGAN [33]	0.671	0.547	0.529	0.545	0.651	0.603	0.585	**0.625**	0.758	0.665	0.618
DSVDD [27]	0.617	**0.659**	0.508	**0.591**	0.609	**0.657**	0.677	**0.673**	0.759	**0.731**	0.648
OCGAN [22]	0.757	0.531	0.640	**0.620**	0.723	**0.620**	0.723	0.575	**0.820**	0.554	**0.657**
GradCon	**0.760**	**0.598**	0.648	0.586	0.733	0.603	0.684	0.567	**0.784**	**0.678**	**0.664**

AUROC results in MNIST

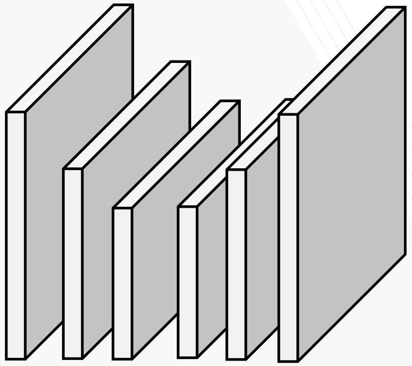
	0	1	2	3	4	5	6	7	8	9	Average
OCSVM [34]	0.988	**0.999**	0.902	0.950	0.955	0.968	0.978	0.965	0.853	0.955	0.951
KDE [4]	0.885	0.996	0.710	0.693	0.844	0.776	0.861	0.884	0.669	0.825	0.814
DAE [9]	0.894	**0.999**	0.792	0.851	0.888	0.819	0.944	0.922	0.740	0.917	0.877
VAE [12]	**0.997**	**0.999**	0.936	0.959	**0.973**	0.964	**0.993**	0.976	0.923	**0.976**	0.970
PixelCNN [20]	0.531	0.995	0.476	0.517	0.739	0.542	0.592	0.789	0.340	0.662	0.618
LSA [1]	0.993	**0.999**	**0.959**	**0.966**	0.956	0.964	**0.994**	**0.980**	**0.953**	**0.981**	**0.975**
AnoGAN [33]	0.966	0.992	0.850	0.887	0.894	0.883	0.947	0.935	0.849	0.924	0.913
DSVDD [27]	0.980	**0.997**	0.917	0.919	0.949	0.885	0.983	0.946	**0.939**	0.965	0.948
OCGAN [22]	**0.998**	**0.999**	0.942	0.963	**0.975**	**0.980**	0.991	**0.981**	**0.939**	**0.981**	**0.975**
GradCon	0.995	**0.999**	**0.952**	**0.973**	0.969	**0.977**	**0.994**	0.979	0.919	0.973	**0.973**

Fashion-MNIST

% of outlier		10	20	30	40	50
F1	GPND	**0.968**	**0.945**	0.917	0.891	0.864
	Grad	0.964	0.939	0.917	0.899	0.870
	GradCon	0.967	**0.945**	**0.924**	**0.905**	**0.871**
AUC	GPND	0.928	0.932	0.933	0.933	0.933
	Grad	0.931	0.925	0.926	0.928	0.926
	GradCon	**0.938**	**0.933**	**0.935**	**0.936**	**0.934**

Computational Efficiency
Inference Time, Model Parameters

GradCon



Covolutional autoencoder

Does not require

- ❌ Adversarial training
- ❌ Autoregressive models

↓ Model parameters

Computations

Average inference time per image for GradCon

(3.08*ms*) is 1.9 times faster than GPND[1] (5.72*ms*)

Method	# of parameters
AnoGAN	6,338,176
GPND	6,766,243
LSA	13,690,160
GradCon	**230,721**

→ Model parameters are

at least 27 time less

[1] NeurIPS 2018

Conclusion

- We propose using a gradient-based representation for anomaly detection by characterizing model behavior on anomalies

- The proposed anomaly detection algorithm, GradCon, achieves state-of-the-art performance with significantly less number of model parameters

- Using training strategies such as adversarial training or probabilistic modeling on gradient-based representations remains for future works

Thanks for your attention

Website

Paper

Code

[Website]: https://ghassanalregib.info/

[Paper]: https://arxiv.org/abs/2007.09507

[Code]: https://github.com/olivesgatech/gradcon-anomaly