Joint Energy-based Model for Remote Sensing Image Processing

Daniel Orozco

Helmholtz AI Conference 2023

Hamburg, 13.06.2023

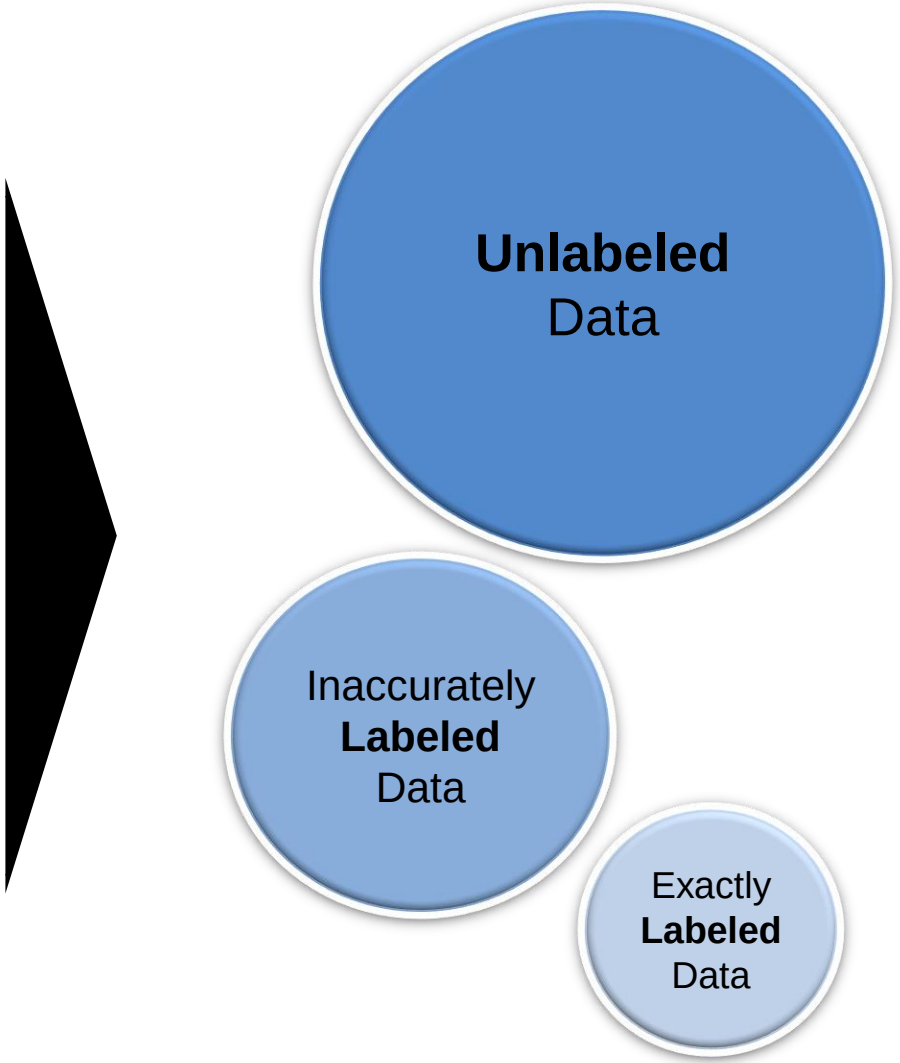
Content

1. Introduction

Earth Observation Data

– **Volume**
  - ESA Sentinel satellites: ~20TiB daily user-level data.[1]
  - Deep Learning models with millions of parameters.

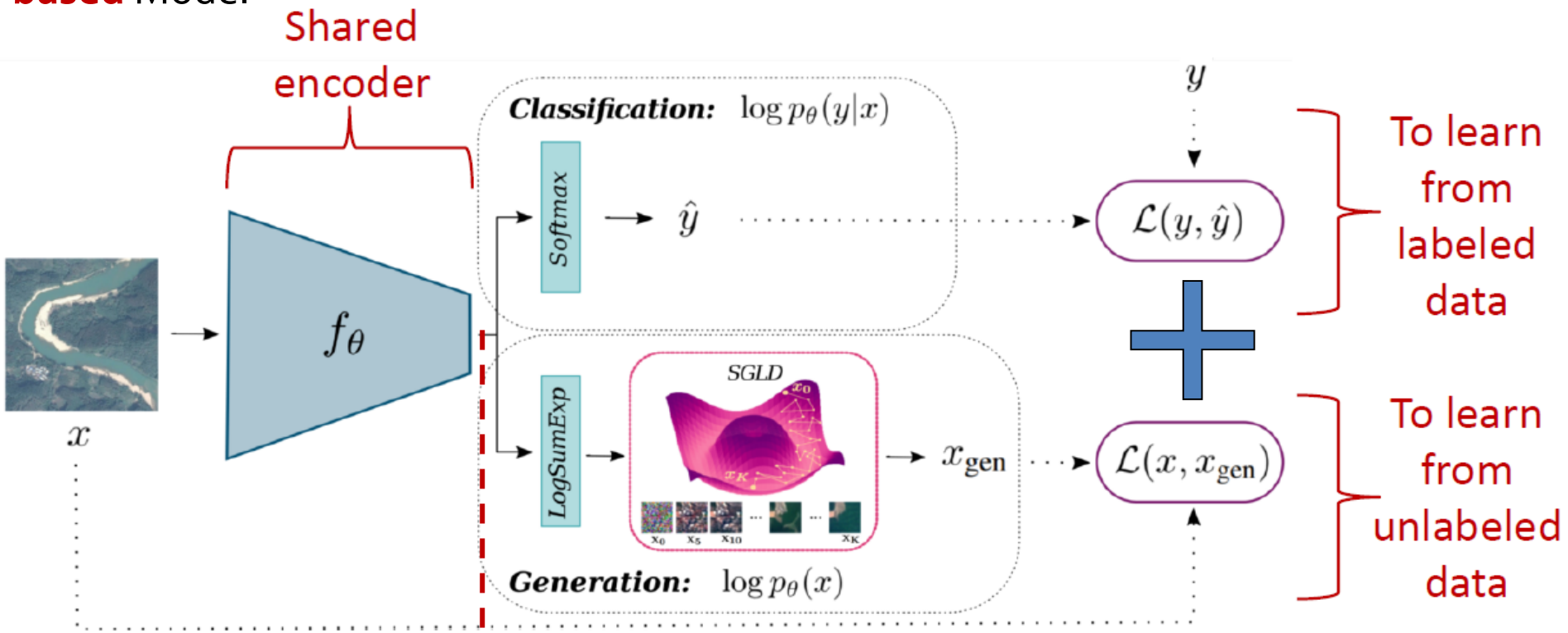
– **Quality**
  - Completely labeled sets are costly.
  - Require a long time to update.
  - High-resolution semantic segmentation maps.



1. Serco Europe, RP. (2021). Sentinel data access annual report Y2020. Copernicus.

2. JEM

Joint Energy-based Model



[3] J. Castillo-Navarro, et al.

Energy function: $E_\theta(x) = -log \sum_y exp(f_\theta(x)[y])$

2.1 Energy function

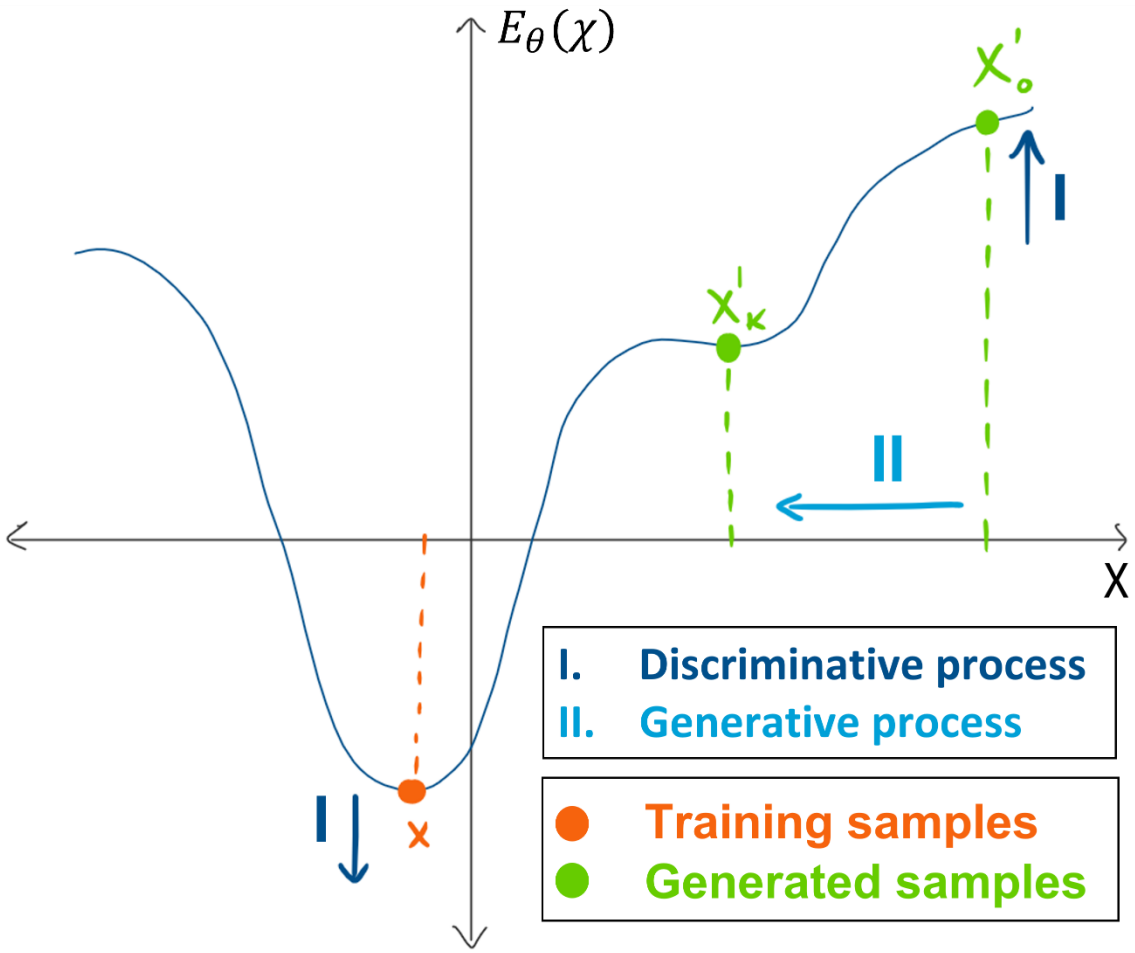
Joint **Energy-based** Model

– Energy function

↑ High values: unrealistic samples.

↓ Low values: realistic samples.

Dual optimization problem



3. JEM in Remote Sensing

Datasets

– **EuroSAT (RGB)**
  - 22,000 (train) and 5,000 (validation & test).
  - 10 classes.

[7] Davies

– **New York City (RGB + NIR)**
  - 6,000 (train) and 1,140 (validation & test).
  - 3 classes.
    - **Buildings**, **Vegetation** and **Bare land**.

Optical

Ground Truth

3. JEM in Remote Sensing

RQ1: **What factors influence the <u>instability</u> of JEM training for Classification and how to alleviate it?**

3.1 JEM training instability

– The training loss ($\mathcal{L}$) indicates how well the model fits the training data.

– JEM training simultaneously optimizes the classification and generation branch:

$$\mathcal{L}_{TOTAL} = \mathcal{L}_{CLA} + \mathcal{L}_{GEN}$$

$$\mathcal{L}_{CLA} = CrossEntropy(y, \hat{y}) = -\sum_{k} y[k] \cdot \log(\hat{y}[k])$$ ➡ **Bounded above 0**

$$\mathcal{L}_{GEN} = E_\theta(x) - E_\theta(x')$$ ➡ **Unbounded**

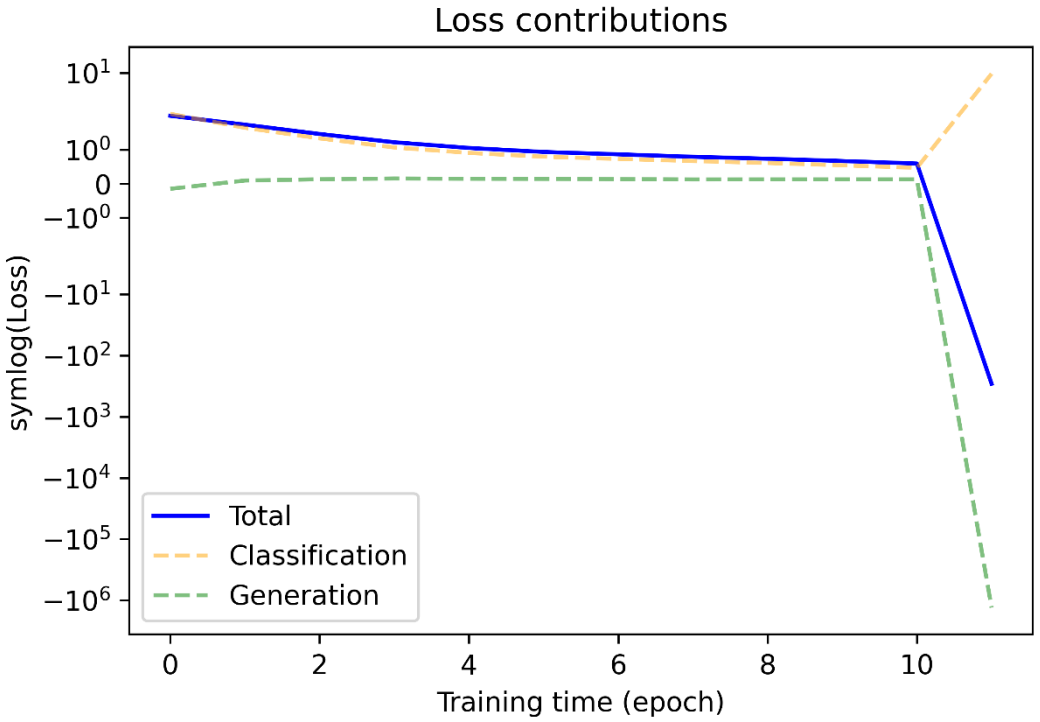
– Model **divergence** due to unbounded definition of generation loss.

3.1 JEM training instability

– Model **divergence** due to unbounded definition of generation loss

Dataset	Completed epochs	Completed epochs (%)
CIFAR-10	51	25.5%
EuroSat	10	5%

Table 3.2. Completed epochs for JEM classification with different datasets.



3.1 Regularization

– **Regularization** adds a penalty term to the objective function.

– Applied to generation loss to **constrain** the values:

$$\mathcal{L}_{GEN\_REG} = \mathcal{L}_{GEN} + \alpha \cdot REG$$

– Three types of **L2** regularization:

$$REG_r = \sum E_\theta(x)^2$$ (Energy values of training/real samples)

$$REG_u = \sum E_\theta(x')^2$$ (Energy values of generated/unreal samples)

$$REG_g = \sum \nabla E_\theta(x)(x)^2$$ (Energy gradients of training/real samples)

3.1 Regularization

– **Regularization** results

Regularization		Completed epochs		Best classification accuracy (validation)
Weight	Type	#	%	
0.2	Energy of training samples REG_r	200	**100%**	**95.16%**
	Energy of generated samples REG_u	155	77.5%	63.42%
	∇Energy of training samples REG_g	188	59%	89.82%

Table 3.5. L2 regularization on energy values of training and generated samples and on energy gradients of training samples.

Regularization type	Enables more training epochs	Complete training	Average time per epoch (seconds)
Energy of training samples REG_r		✓	288
Energy of generated samples REG_u	✓	✗	289
∇Energy of training samples REG_g			398

Table 3.6. Regularization types implications.

3.1 Regularization

– **Regularization and hyper-parameters** results

Learning rate	Regularization		Completed epochs		Best classification accuracy (validation)
	Type	Weight	#	%	
1×10^{-4}	Energy of training samples	0	10	5%	84%
		0.5	7	3.5%	54%
		1	6	3%	40%
1×10^{-5}	Energy of training samples	0	86	43%	88.6%
		0.2	200	**100%**	**95.16%**
		0.5	31	15.5%	66.8%
		0.75	30	15%	63%
	Energy of generated samples	0.2	155	77.5%	63.42%
	∇Energy of training samples	0.2	118	59%	89.82%

Table 3.10. Completed epochs for different hyper-parameters and regularizations.

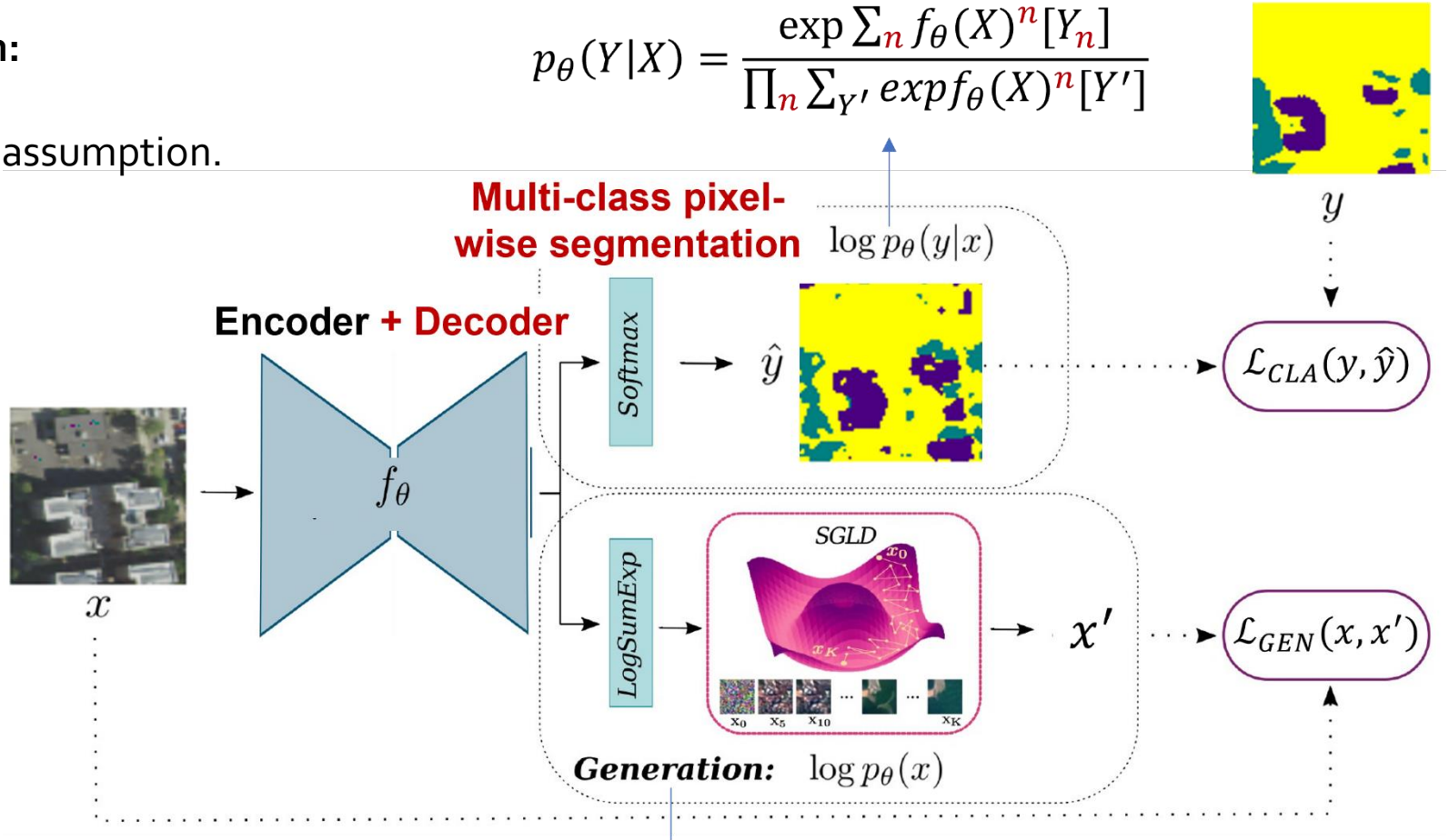
3. JEM in Remote Sensing

RQ2: **What are the potential and challenges of JEM for <u>semantic segmentation</u> of remote sensing data?**

3.2 JEM for Segmentation (sJEM)

– **Architecture adaptation:**
  - Decoder.
  - Pixel independence assumption.

$$p_\theta(Y|X) = \frac{\exp \sum_n f_\theta(X)^n[Y_n]}{\prod_n \sum_{Y'} exp f_\theta(X)^n[Y']}$$



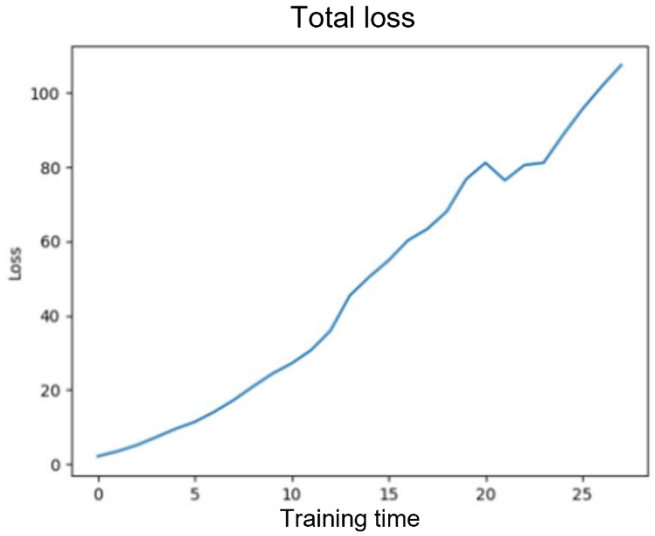
$$E_\theta(x) = -log \prod_n \sum_Y exp(f_\theta(X)^n[Y])$$

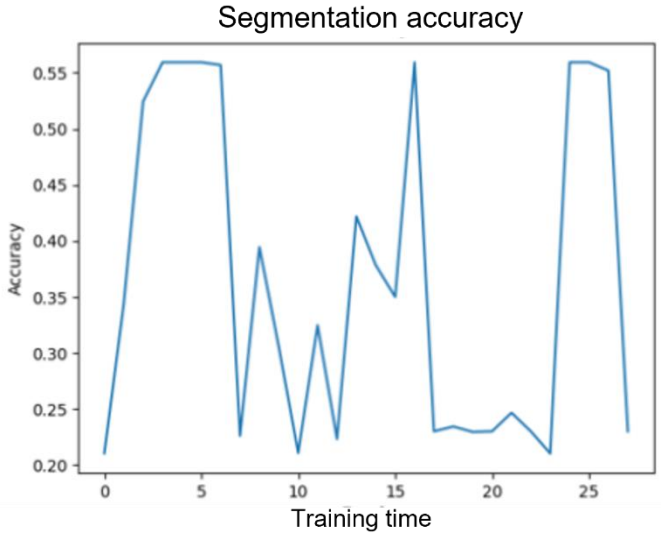
3.2 JEM for Segmentation (sJEM)

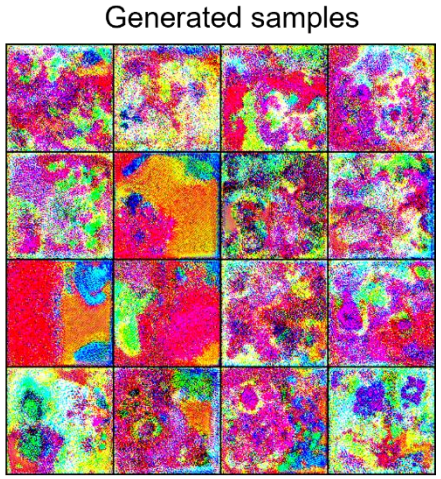
– **Vanilla U-Net** results

Architecture	Training size (samples)	Validation size (samples)	Sample size (pixels)	Best segmentation accuracy (validation)
Vanilla U-Net	6,000	1,140	256x256	29.67%
			128x128	55.94%
			64x64	**56.62%**

Table 4.2. Segmentation results with U-Net and different sample sizes.







3.2 JEM for Segmentation (sJEM)

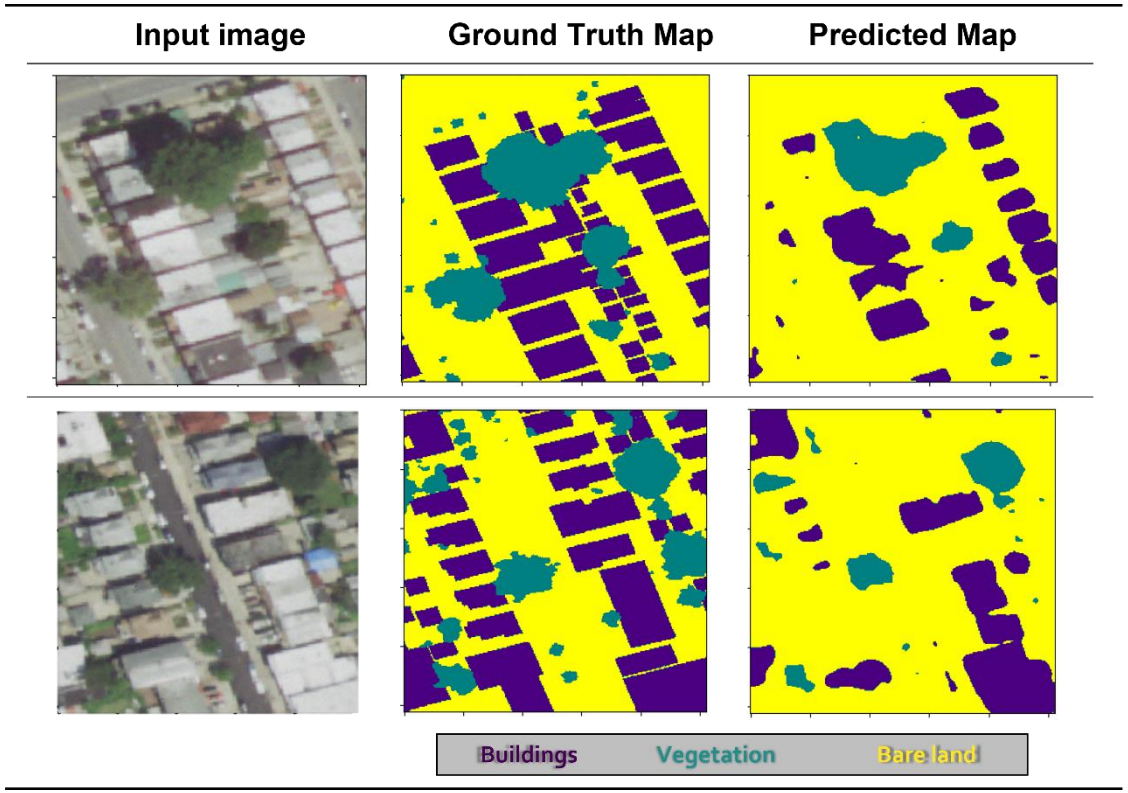
– **Wide-ResNet + U-Net** results:
  - +10% accuracy w.r.t. Vanilla U-Net architecture.

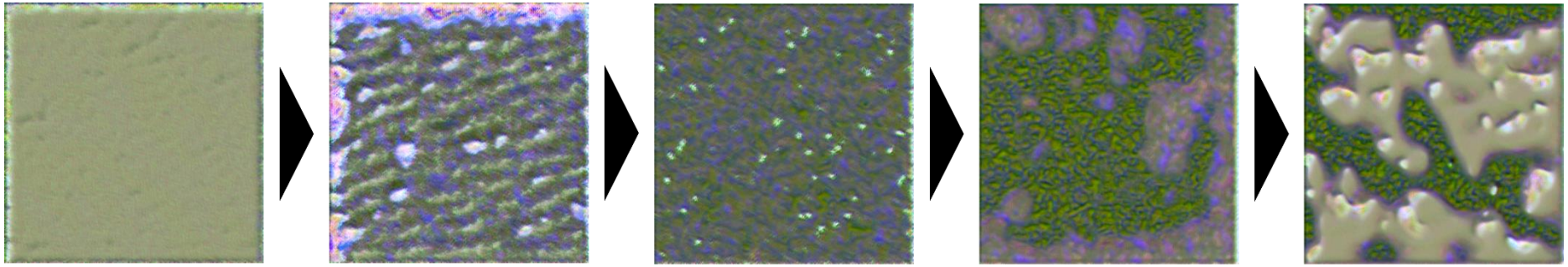
Architecture	Sample size (pixels)	Training size (samples)	Validation size (samples)	Best segmentation accuracy (validation)
Wide-ResNet		10	5	**67.88%**
+	128x128	80	20	62.43%
U-Net		2,000	20	47.50%

Table 4.8. Segmentation results with Wide-ResNet + U-Net and different sizes for training and validation sets.

3.2 JEM for Segmentation (sJEM)

– **Wide-ResNet + U-Net** results

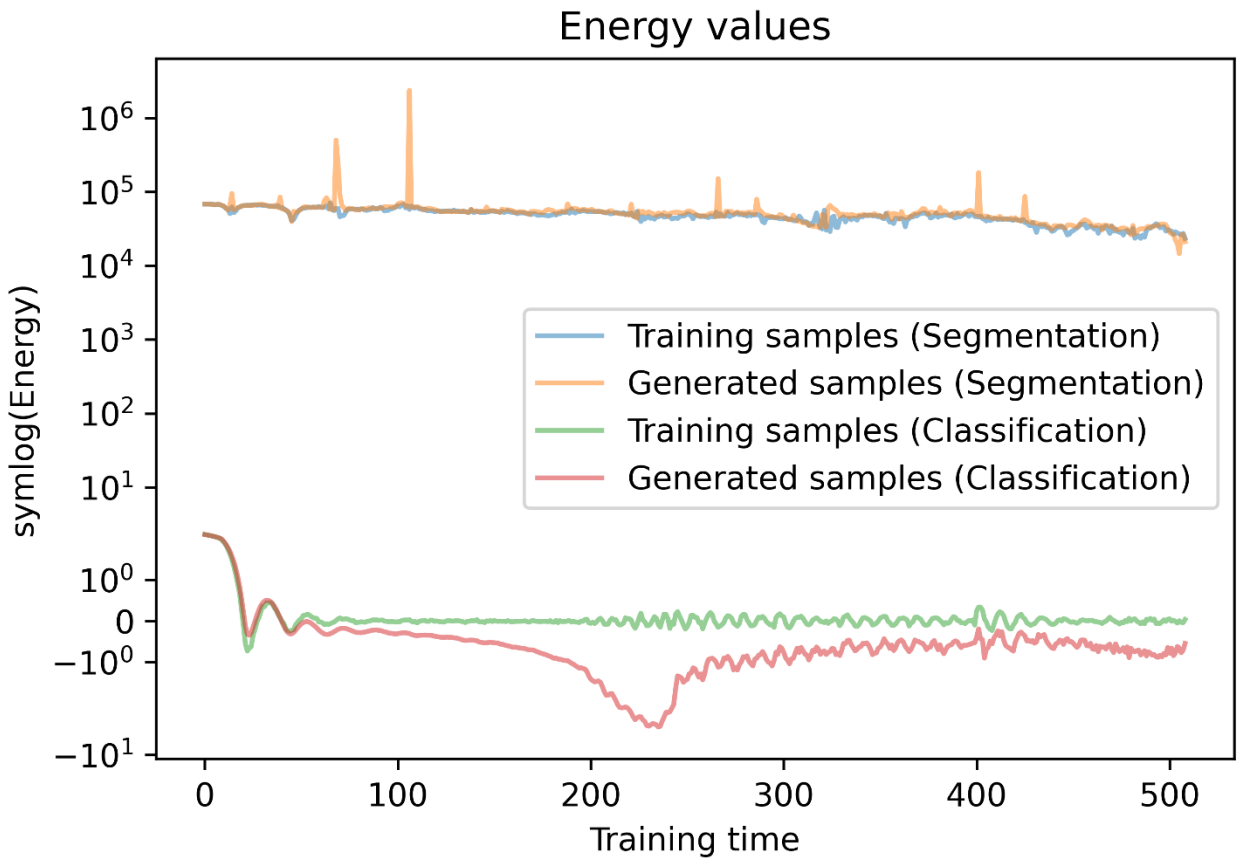




Epoch 1 Epoch 3 Epoch 6 Epoch 9 Epoch 15

3.2 JEM for Segmentation (sJEM)

– **Divergence**
- Pixel aggregations produce energy values ~4 orders of magnitude bigger w.r.t. classification.



4. Conclusions and future work

Divergence:

– Regularization stabilizes the training and alleviates the divergence to some extent.
– Regularization eases the competition of the dual optimization problem.

sJEM:

– The pixel independence assumption is too strong for RS images where spatial autocorrelation is a common characteristic.
– Stronger regularization techniques are required for RS image segmentation.

Future work:

– Further regularization terms that are more effective for divergence.
– Robust definitions to aggregate pixels' energy values, including measures of spatial autocorrelation.
– Feedback sensibility in the generation process to "bad-looking" samples.

Thanks for your attention!

Questions?

References

[1] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, « Your classifier is secretly an energy-based model and you should treat it like one », Proceedings of the International Conference on Learning Representations (ICLR), 2020.

[2] S. Zhao, J.-H. Jacobsen, and W. Grathwohl, « Joint energy-based models for semi-supervised classification », in Proccedings of the International Conference on Machine Learning - Workshop on Uncertainty and Robustness in Deep Learning (ICMLW), 2020.

[3] J. Castillo-Navarro, B. Le Saux, A. Boulch, and S. Lefèvre, « Energy-based models in Earth observation: from generation to semi-supervised learning », IEEE Transactions on Geoscience and Remote Sensing, 2021.

References

[4] Serco Europe, RP. (2021). Sentinel data access annual report Y2020. Copernicus. https://scihub.copernicus.eu/twiki/pub/SciHubWebPortal/AnnualReport2020/COPE-SERCO-RP-21-1141_-_Sentinel_Data_Access_Annual_Report_Y2020_final_v2.3.pdf

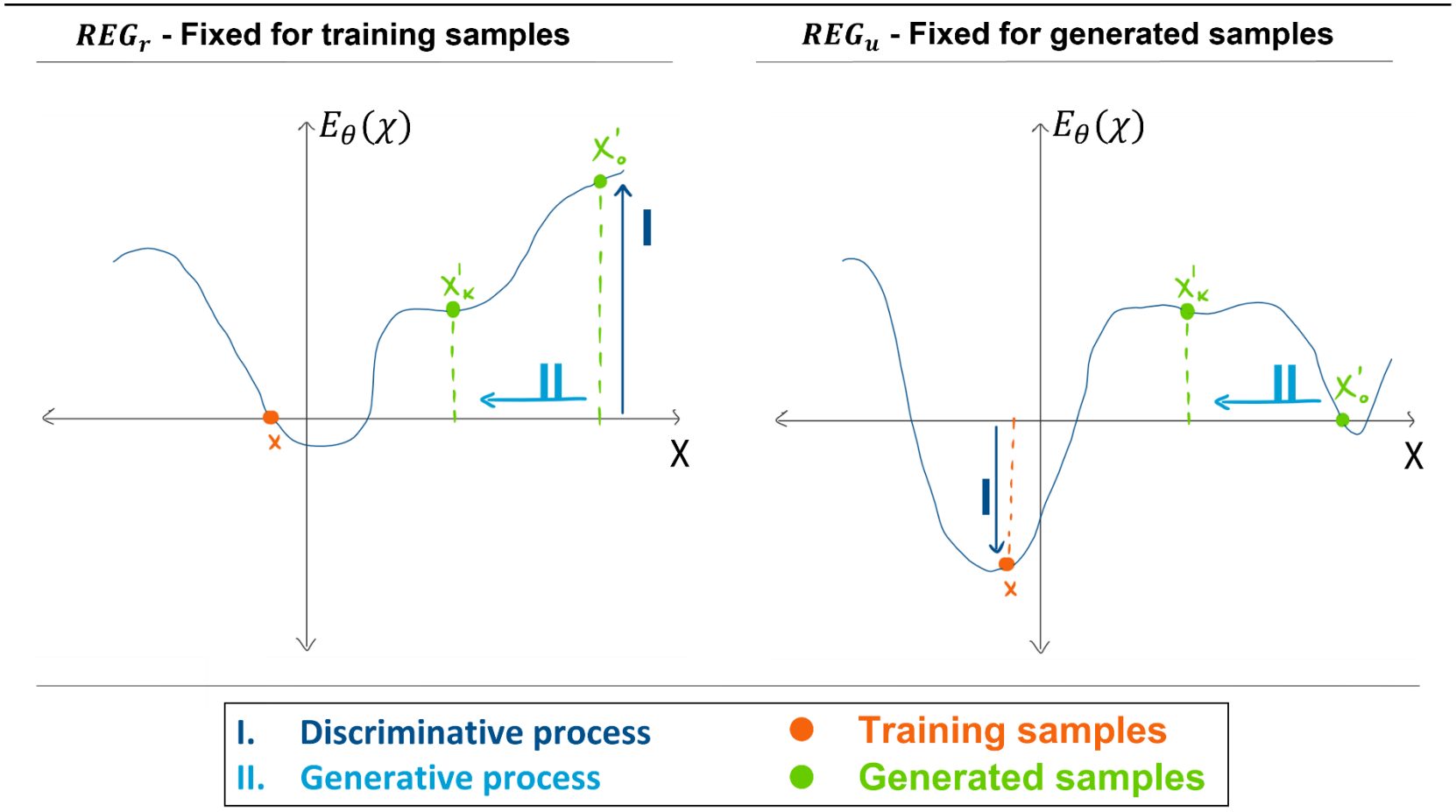
[5] P. Helber, B. Bischke, A. Dengel, and D. Borth. "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification." In: IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 12.7 (2019), pp. 2217–2226.

[6] C. M. Albrecht, F. Marianno, and L. J. Klein. "Autogeolabel: Automated label generation for geospatial machine learning." In: 2021 IEEE International Conference on Big Data (Big Data). IEEE. 2021, pp. 1779–1786.

[7] Davies. A, « Computer Vision | Land Cover Classification Using TensorFlow in Python », in Towards Data Science, 2022.

Appendix | JEM training instability

– **Regularization** in the view of energy landscape



Appendix | JEM training instability

– Influence of **hyperparameters**: learning rate

Learning rate	Completed epochs	Completed epochs (%)	Best classification accuracy (validation)
1×10^{-4}	10	5%	84%
1×10^{-5}	86	**43%**	**88.6%**

Table 3.8. Completed epochs for different learning rates.

Appendix | JEM training instability

– Influence of **hyperparameters**: SGLD steps

Number of SGLD steps	Average time per epoch (seconds)	Completed epochs	Completed epochs (%)
40	2310	10	5%
100	5080	9	4.5%

Table 3.9. Completed epochs and average time per epoch for different number of SGLD steps.

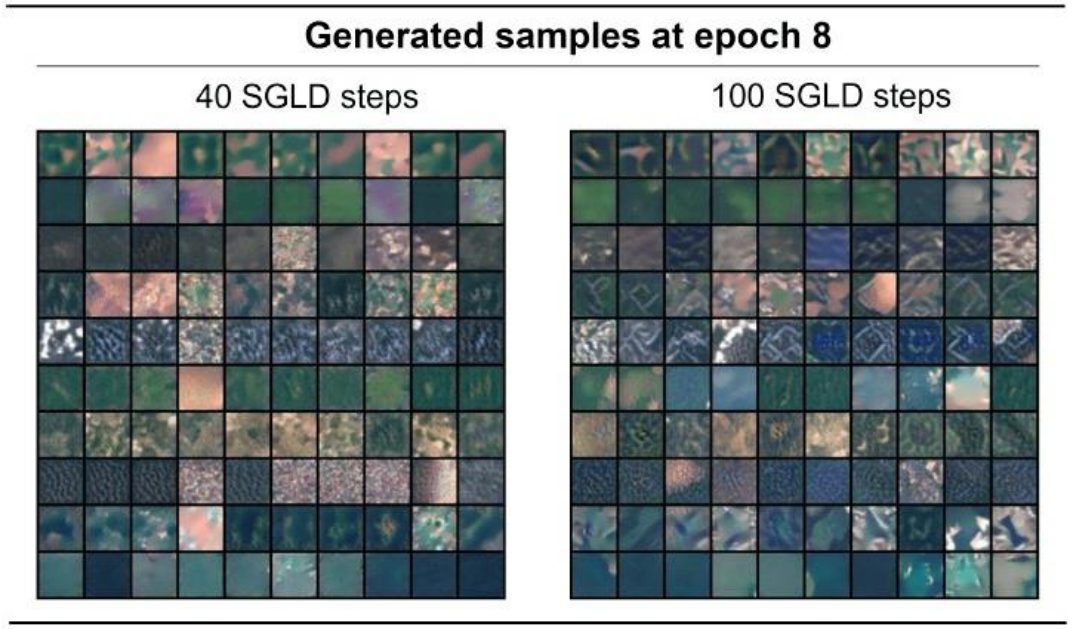
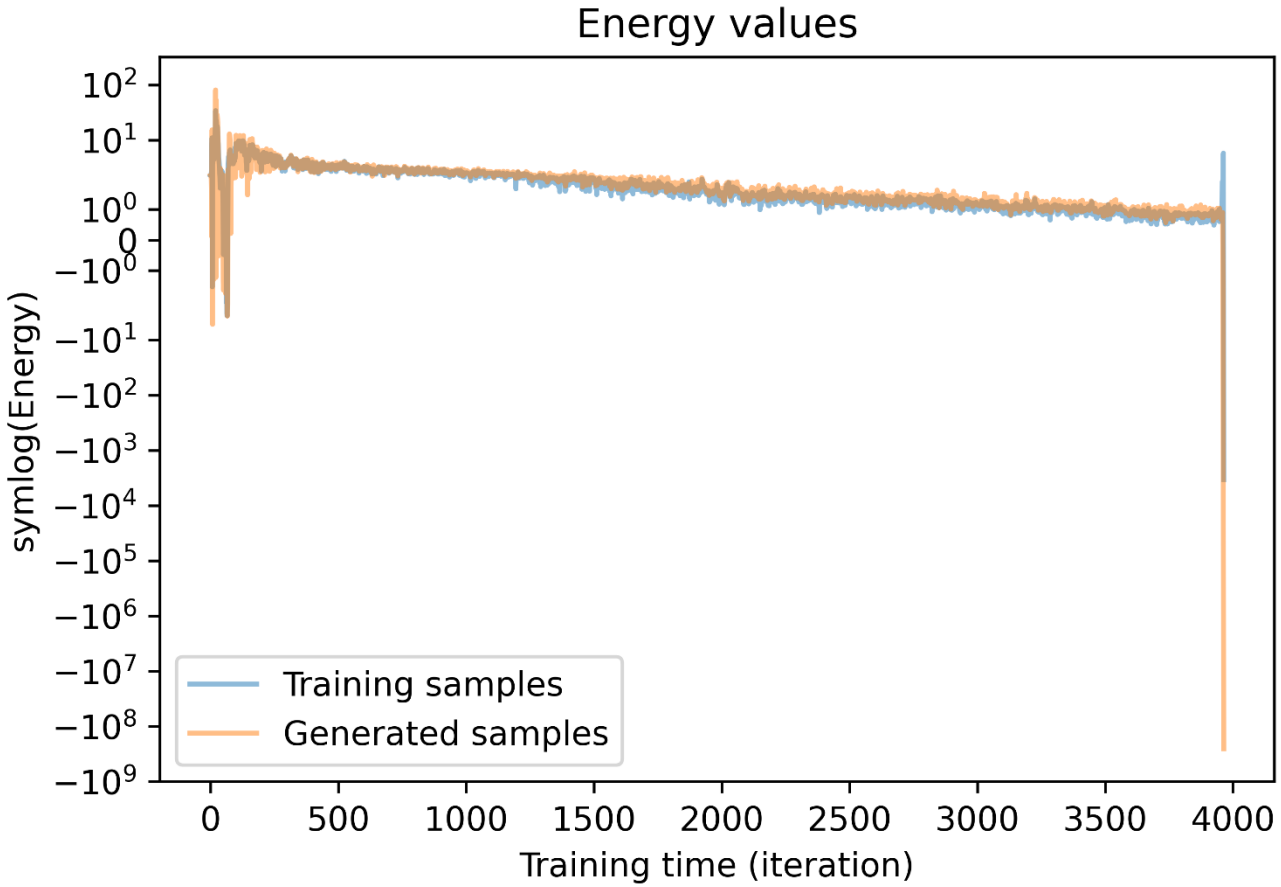


Figure 3.11. Generated samples for different number of SGLD steps.

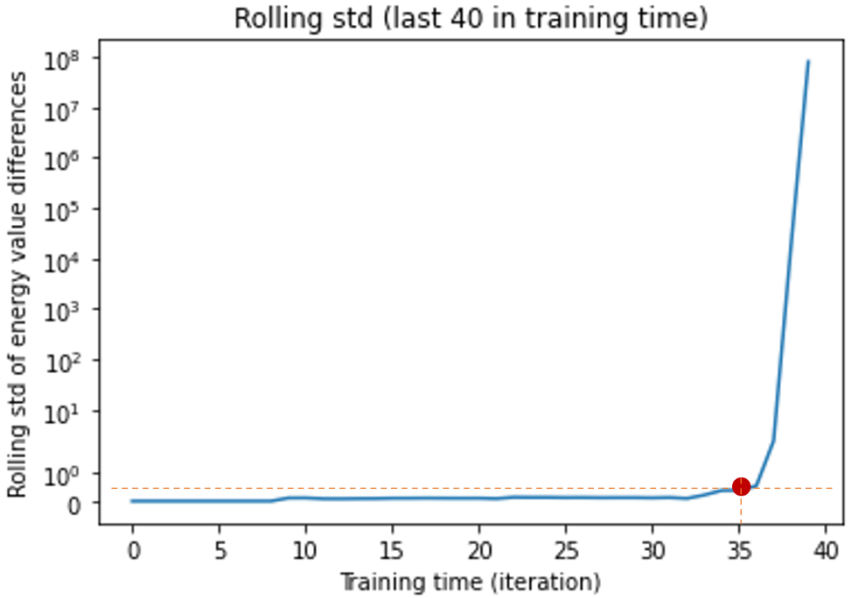
Appendix | JEM training instability

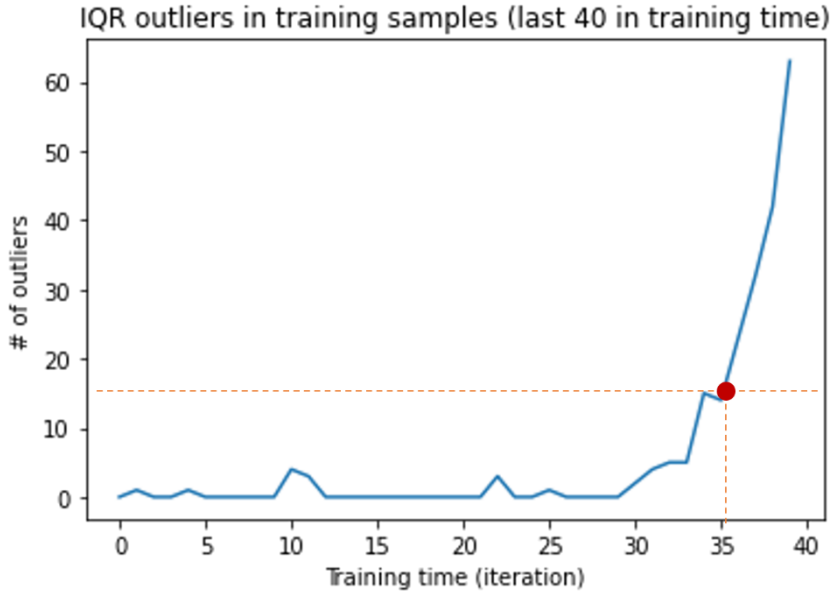
– **Detection** and **Restart**



Appendix | JEM training instability

– **Detection** and **Restart**



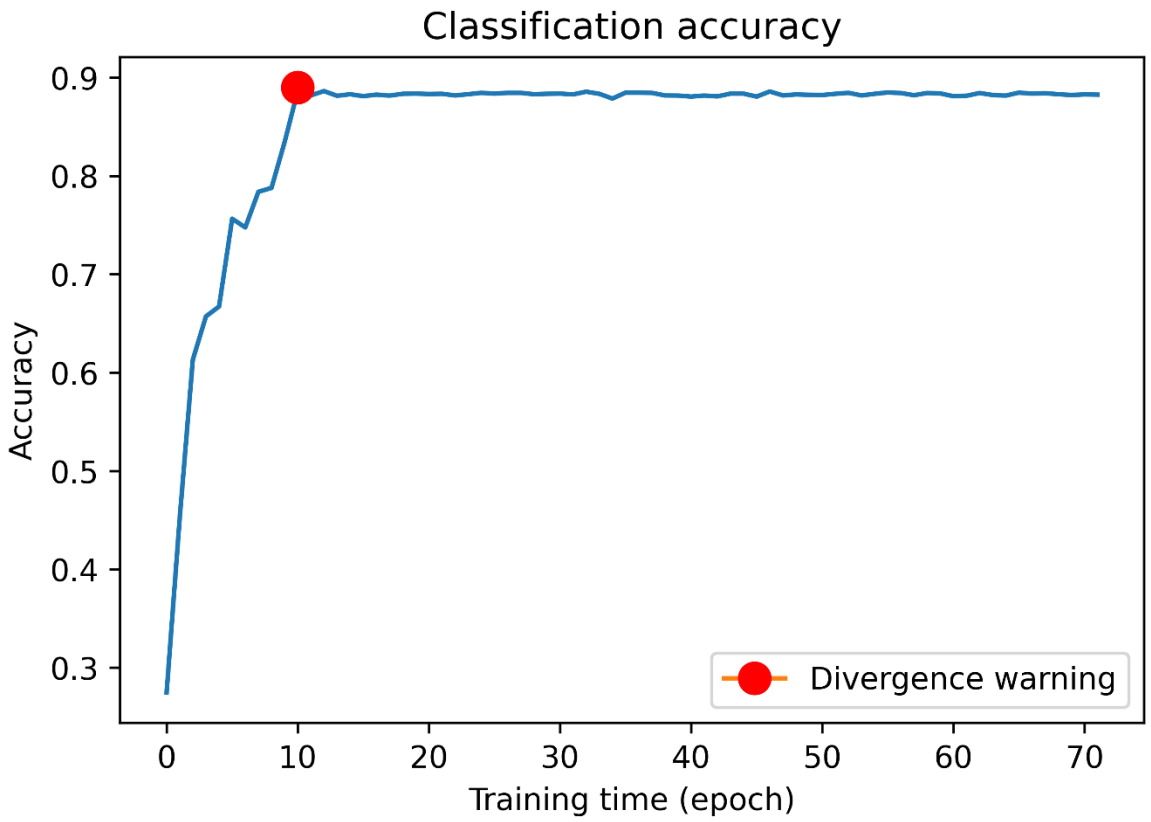


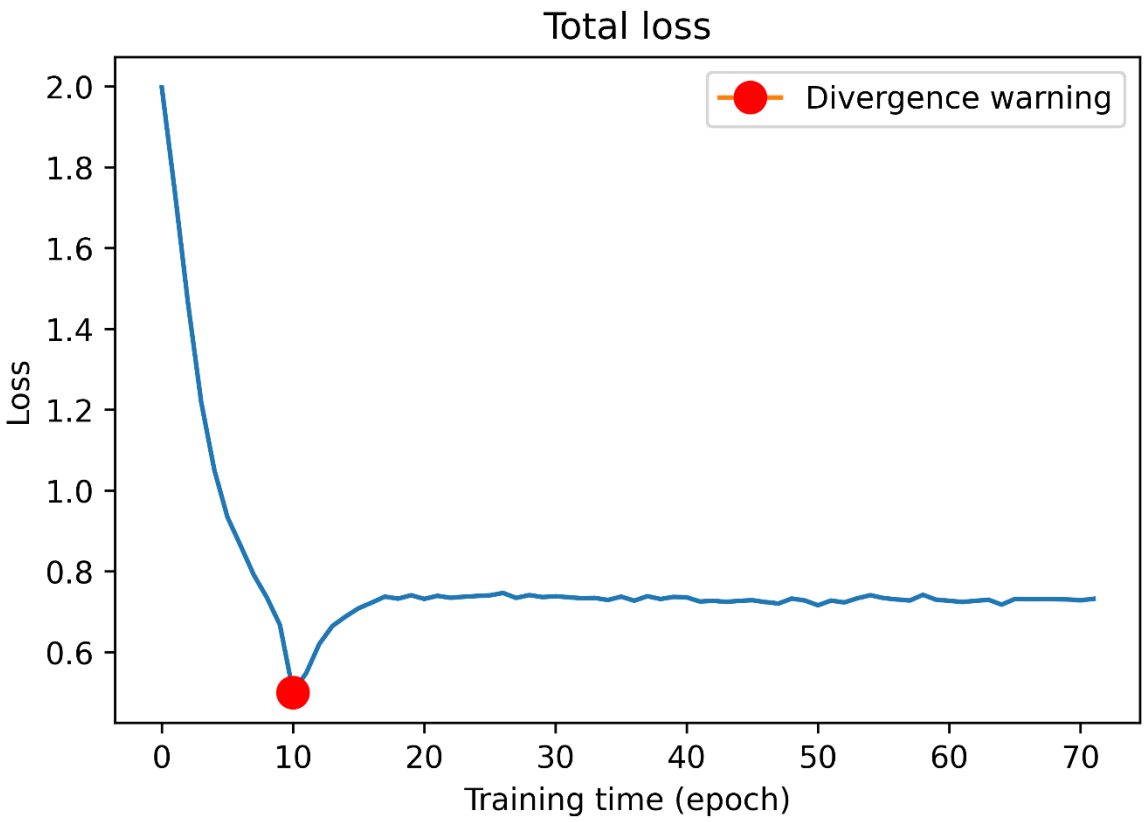
Threshold	Value
Rolling standard deviation α_{RSD}	0.3
Number of outliers α_{OTL}	15

Table 3.3. Thresholds for divergence detection.

Appendix | JEM training instability

– **Detection** and **Restart**





Appendix | JEM training instability

– **Detection** and **Restart**

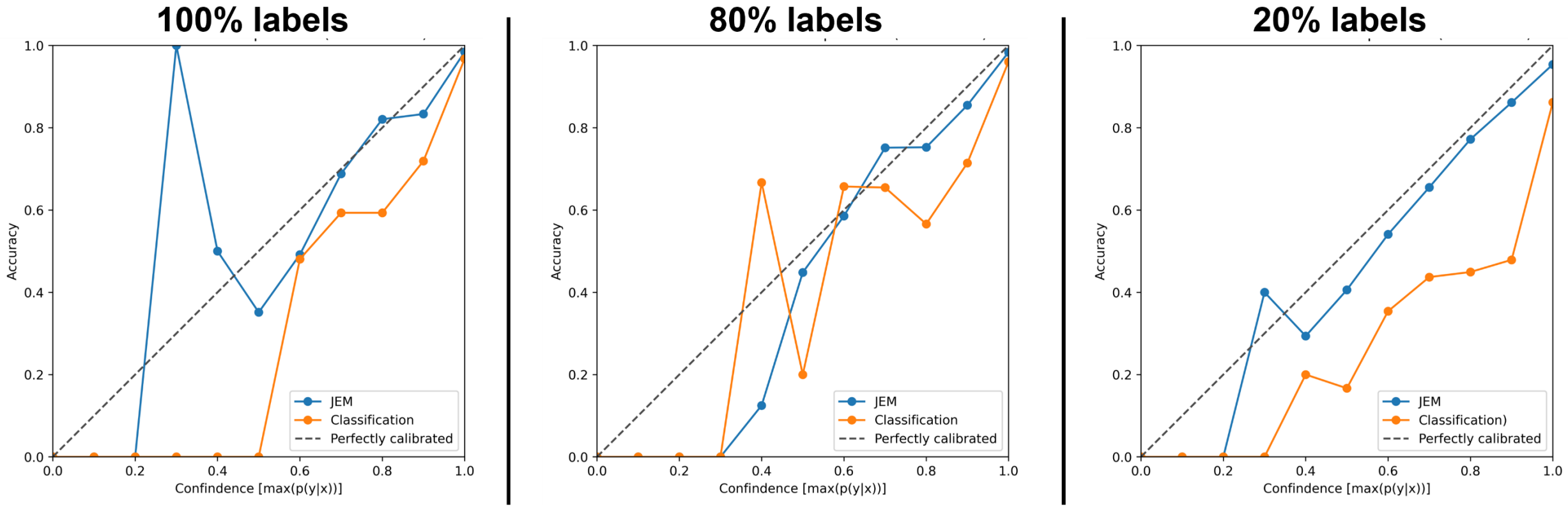
Algorithm 1 Pre-emptive detection of divergence and restart of training

$\alpha_{RSD} \leftarrow 0.3$
$\alpha_{OTL} \leftarrow 15$
$jump\_size \leftarrow 30$
$\omega \leftarrow 10$ ▷ window size for the rolling standard deviation

$current\_rstd \leftarrow RollingStd(|x - x'|, \omega)$
$num\_outliers \leftarrow \sum_x IQR\_Outlier(x)$
if $(current\_rstd \geq \alpha_{RSD})$ & $(num\_outliers \geq \alpha_{OTL})$ **then**
  Divergence warning!
  $previous\_replay\_buffer \leftarrow LoadBuffer(jump\_size)$
  $previous\_optimizer \leftarrow LoadOptimizer(jump\_size)$
  $previous\_model \leftarrow LoadModel(jump\_size)$
  $RestartTraining(previous\_replay\_buffer, previous\_optimizer, previous\_model)$
end if

Appendix | JEM for Classification

– Potential for **semi-supervised learning**



Appendix | JEM for Classification

– Potential for **semi-supervised learning**

Mode (% labels)	Configuration	Mean accuracy	ECE
Supervised (100% labels)	JEM	95.16%	1.08%
	Classification	94.86%	3.37%
Semi-supervised (80% labels)	JEM	94.06%	0.71%
	Classification	94.42%	4.10%
Semi-supervised (20% labels)	JEM	80.76%	0.91%
	Classification	81.02%	15.05%

Table 3.13. EuroSat supervised vs semi-supervised classification accuracy and ECE.

Appendix | JEM for Segmentation (sJEM)

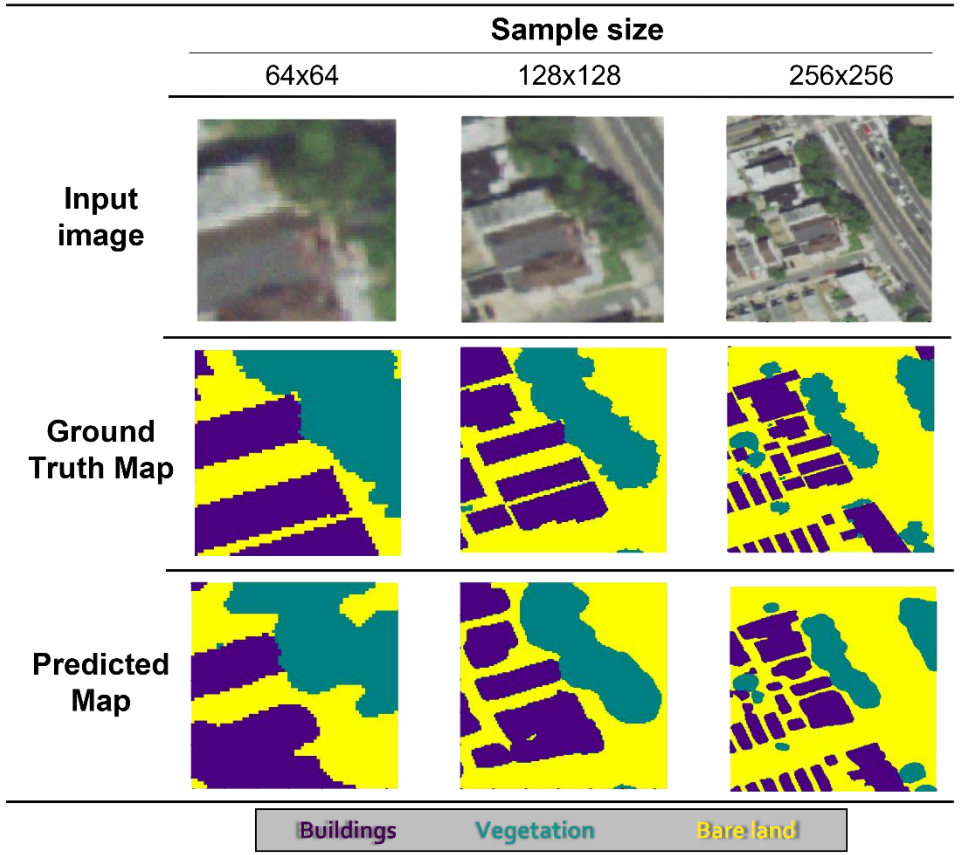
– Individual **segmentation** branch analysis:
  - U-Net architecture.
  - Balance between training time and accuracy.
  - Dice metric addition to segmentation loss.

Sample size (pixels)	Average time per epoch (seconds)	Best segmentation accuracy (validation)
64x64	11	83.83%
128x128	21	86.91%
256x256	54	88.67%

Table 4.3. Segmentation branch results with Vanilla U-Net and different sample sizes.

Sample size (pixels)	Segmentation Loss	Best segmentation accuracy (validation)
64x64	Cross-Entropy	83.83%
	Cross-Entropy + Dice	84.40%
128x128	Cross-Entropy	86.91%
	Cross-Entropy + Dice	87.05%
256x256	Cross-Entropy	88.67%
	Cross-Entropy + Dice	88.72%

Table 4.4. Segmentation branch results with different loss metrics.



Appendix | JEM for Segmentation (sJEM)

– Individual **generation** branch analysis:
  - 1 sample.
  - Different encoders and learning rates.

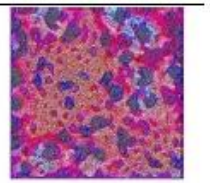
Training sample	Encoder	Learning rate	Generated sample
	ResNet18/34/50	1×10^{-2}	
		1×10^{-3}	
		1×10^{-4}	
		1×10^{-5}	
	U-Net	1×10^{-3}	
		1×10^{-4}	
		1×10^{-5}	
	Wide-ResNet	1×10^{-3}	
		1×10^{-4}	

Table 4.6. Generation branch results with different encoders and learning rates.

Appendix | JEM for Segmentation (sJEM)

– Influence of **hyperparameters**: learning rate and buffer size

Learning rate	Buffer size	Best segmentation accuracy (validation)
1×10^{-4}	3,000	**67.88%**
	6,000	59.48%
	9,000	39.25%
	30,000	38.26%
	60,000	38.37%
1×10^{-5}	3,000	39.02%
	6,000	38.86%

Table 4.7. Segmentation results with Wide-ResNet + U-Net, different learning rates, and buffer sizes.