Pre-trained Encoders in Self-Supervised Learning Improve Secure and Privacy-preserving Supervised Learning

Hongbin Liu, Wenjie Qu, Jinyuan JiaNeil, Zhenqiang Gong

Use pre-trained encoders from self-supervised learning

Problem: Supervised learning has security and privacy vulnerabilities, such as adversarial attacks and inference attacks.

Solution: Use pre-trained encoders from self-supervised learning to improve accuracy, security, and privacy in supervised tasks.

Findings:

- Improved testing accuracy and certified security guarantees.
- Better efficiency and accuracy for privacy-preserving tasks like machine unlearning and differential privacy.

Introduction

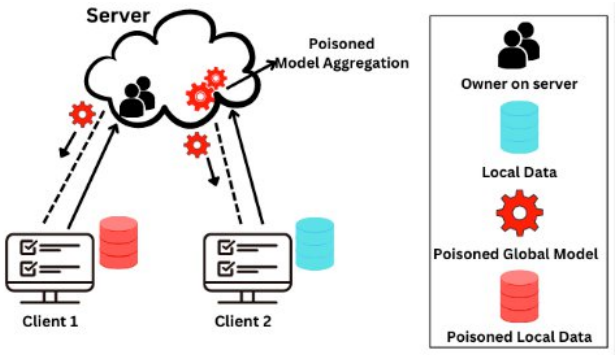
Supervised Learning

- Used in critical applications like cybersecurity and healthcare.
- Relies on labeled data but faces security issues like data poisoning and backdoor attacks.
- Privacy challenges include compliance with GDPR and protection against inference attacks.

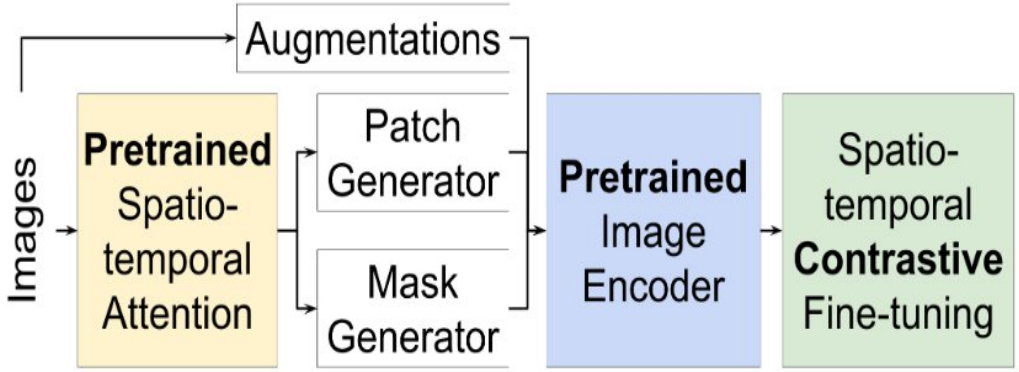
Self-Supervised Learning

- Pre-training encoders on unlabeled data extracts high-quality features.
- Enables building accurate classifiers with minimal labeled data.

Data Poisoning



Pre-training encoders



Limitations in Supervised Learning

Security Limitations in Supervised Learning

- Reduced accuracy when using secure algorithms (e.g., Bagging, KNN).
- Weak certified security guarantees.

Privacy Limitations in Supervised Learning

- Differentially private methods reduce testing accuracy.
- Exact machine unlearning is inefficient.

Self-Supervised Learning

- Pre-trained encoders learn general features from large, unlabeled datasets.
  Example: OpenAI's CLIP trained on 400M image-text pairs.

- Encoders extract better features, improving downstream supervised learning tasks.
  Public data removes privacy concerns.

- Two Approaches for Training:
    1. **Linear Probing (LP):** Train only the classifier layer.
    2. **Fine-Tuning (FT):** Update the encoder and classifier layers.

Measurement Setup

Datasets: STL10, CIFAR10, Tiny-ImageNet.

	STL10	CIFAR10	Tiny-ImageNet
#Classes	10	10	200
#Training examples	5,000	50,000	100,000
#Validation examples	4,000	5,000	10,000
#Testing examples	4,000	5,000	10,000

Case I: Simple linear classifier.
Case II: Deep neural network (DNN) trained from scratch.
Case III: Pre-trained encoder with LP or FT.

	STL10	CIFAR10	Tiny-ImageNet
Case I	0.353	0.438	0.069
Case II	0.755	0.953	0.516
Case III-LP	0.985	0.954	0.751
Case III-FT	0.970	0.973	0.785

Metrics:

1. Testing accuracy.
2. Certified security guarantees.
3. Efficiency and accuracy for privacy-preserving tasks.

Security Improvement

Improved Accuracy:

- Case III with pre-trained encoders achieves higher accuracy across datasets (e.g., STL10: 97.9%).

Against Data Poisoning/Backdoor Attacks:

- Pre-trained encoders improve certified poisoning size and backdoor attack resistance (e.g., Bagging's ACPS increases significantly in Case III).

Randomized Smoothing:

- Larger certified radii without accuracy loss.

Comparing cases between bagging and knn

(a) Testing accuracy under no attacks

	STL10	CIFAR10	Tiny-ImageNet
Case I	0.352	0.358	0.071
Case II	0.559	0.566	0.072
Case III-LP	0.979	0.907	0.629

(b) ACPS against data poisoning attacks

	STL10	CIFAR10	Tiny-ImageNet
Case I	3.7	21.2	0.001
Case II	5.0	20.7	0.06
Case III-LP	70.3	359.0	19.5

(c) ACPS against backdoor attacks

	STL10	CIFAR10	Tiny-ImageNet
Case I	3.7	20.9	0.001
Case II	5.0	20.3	0.06
Case III-LP	69.9	357.5	19.4

(a) Testing accuracy under no attacks

	STL10	CIFAR10	Tiny-ImageNet
Case I	0.194	0.229	0.029
Case II	0.230	0.327	0.038
Case III	0.963	0.865	0.583

(b) ACPS against data poisoning attacks

	STL10	CIFAR10	Tiny-ImageNet
Case I	36.6	187.8	10.3
Case II	14.3	164.2	13.2
Case III	138.7	929.4	51.6

(c) ACPS against backdoor attacks

	STL10	CIFAR10	Tiny-ImageNet
Case I	36.1	187.3	10.3
Case II	14.3	163.3	13.2
Case III	138.5	962.2	51.3

Privacy Preservation

Differentially Private Classifiers:

- Case III-LP maintains high accuracy under strong privacy constraints (e.g., STL10 accuracy: 95.6%).
- Case III-FT less effective under strong privacy constraints due to noise sensitivity.

Exact Machine Unlearning:

- Retraining from scratch is more efficient and accurate in Case III due to pre-trained features.

Efficiency Analysis

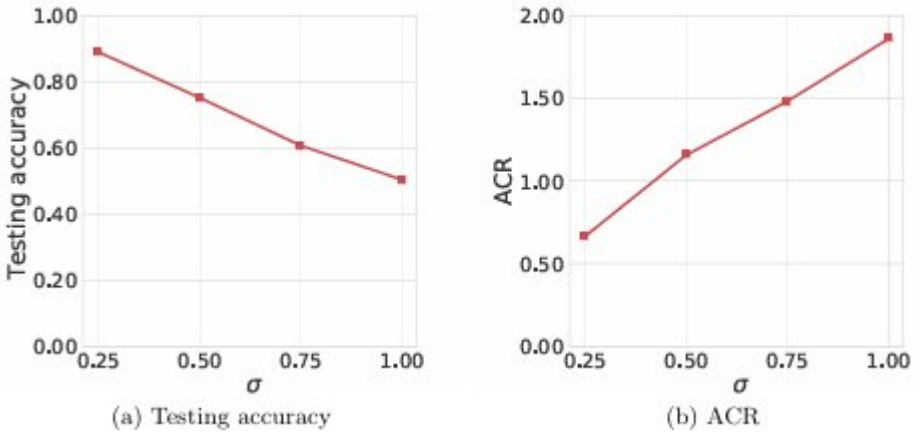
Computation and Storage Costs:

- Linear probing (LP) is more efficient than fine-tuning (FT).
- FT achieves slightly better performance but at a much higher computational cost.

Practical Recommendations:

- Use LP for efficient privacy-preserving and secure learning.

Impact of on the testing accuracy and ACR of randomized smoothing



(a) Testing accuracy (b) ACR

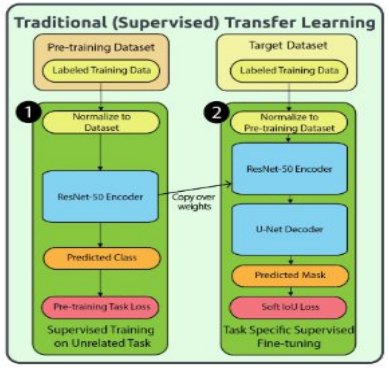
Conclusions

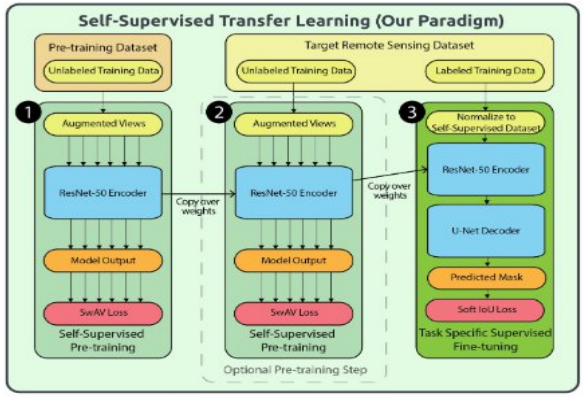
Key Takeaways

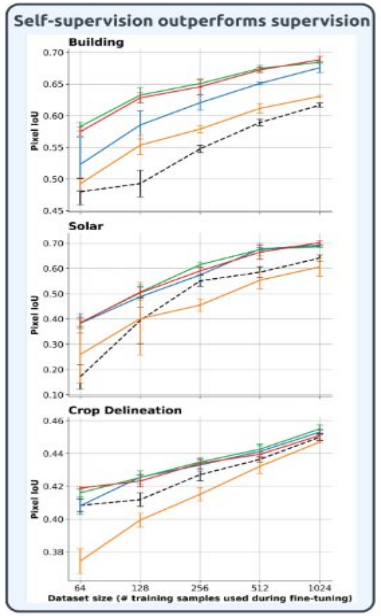
- Pre-trained encoders significantly enhance both security and privacy in supervised learning.
- They mitigate accuracy and efficiency trade-offs in traditional methods.
- Case III with linear probing offers the best trade-off between accuracy, security, and efficiency.

Future Work

- Investigating encoder robustness and privacy-preserving pre-training.







Datasets Used for Experiments

Building dataset: semantic segmentation; segment out buildings

Solar PV: semantic segmentation dataset; segment out rooftop and ground solar photovoltaic arrays.

Crop Delineation: semantic segmentation; segment out borders marking edges of crop fields.

Experimental Design and Graph Legend

Pre-training method	Stage 1 Pre-training Dataset (weight initialization)	Stage 2 Pre-training Dataset
Supervised	ImageNet	None
Self-supervised (SwAV)	ImageNet	None
Self-supervised (SwAV)	None	Target
Self-supervised (SwAV)	ImageNet	Target
Self-supervised (SwAV)	ImageNet	Building + Field Delineation + Solar

References

[2] Public pre-trained image encoder by Google. https://storage.cloud.google.com/simclr gcs/checkpoints/ResNet50 1x.zip, 2021.

[4] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Proceedings of the 2016 ACMSIGSACconference on computer and communications security, pages 308318, 2016.

[5] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3):175185, 1992.

[6] Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. International Journal of Security and Networks, 10(3):137150, 2015