

PREFACE TO THE EDITION

It is a pleasure to present the forthcoming issue of the **Eduschool Journal of Computer Science Research Studies (EJCSRS)**, which brings together a timely collection of scholarly works addressing some of the most significant challenges and emerging directions in contemporary computer science. The contributions in this issue reflect the rapidly evolving nature of intelligent computing, with particular emphasis on security, robustness, privacy, trustworthy artificial intelligence, and scalable distributed technologies.

The issue opens with a comprehensive review of adversarial attacks and certified defences for deep image classifiers, examining the vulnerability of deep learning systems to carefully crafted perturbations and the distinction between empirical robustness and formally certified guarantees. The article compares major certification approaches and highlights the continuing gap between clean and robust accuracy, while emphasizing the importance of adaptive attack evaluation in assessing the reliability of defensive mechanisms.

The second contribution explores diffusion models for image restoration and super resolution, reflecting the growing role of generative models in reconstructing degraded visual information. The study examines conditional models and prior-based approaches across problems such as blur, noise, downsampling, and missing pixels. It also draws attention to the important relationship between distortion-based and perceptual evaluation, as well as unresolved challenges including blind degradation and the generation of hallucinated details.

The issue further extends the application of artificial intelligence to cybersecurity and financial security through graph neural networks for malware and financial fraud detection. By treating malicious software and fraudulent activity as graph-structured phenomena, the contribution demonstrates how structural relationships among program components, accounts, devices, and attributes can be exploited for detection. At the same time, it identifies persistent challenges involving class imbalance, adaptive adversaries, camouflage, and inappropriate temporal evaluation protocols.

Privacy and intellectual-property concerns in deployed artificial intelligence systems are examined in the article on model extraction and membership inference attacks. The contribution distinguishes attacks targeting the reconstruction of proprietary models from those seeking to determine whether particular records were used during model training. It also discusses defensive approaches and highlights the limitations of commonly used evaluation practices and watermarking strategies, reinforcing the need for stronger approaches to protecting both models and the individuals whose data contribute to them.

The issue concludes with an exploration of zero-knowledge proofs in distributed systems, with particular attention to zk-SNARKs and zk-STARKs and their applications in privacy-preserving and scalable blockchain systems. The article presents the fundamental principles behind zero-knowledge proofs and examines the practical trade-offs between proof size, verification efficiency, trusted setup requirements, and security. Its discussion of confidential transactions and validity rollups demonstrates the continuing convergence of cryptography, distributed systems, privacy, and blockchain technology.

Taken together, the contributions in this issue demonstrate an important shift in computer science research: technological advancement must increasingly be accompanied by security, verifiability, privacy, robustness, and responsible evaluation. Whether the subject is an intelligent image classifier, a generative restoration model, a graph-based security system, a deployed machine-learning service, or a decentralized blockchain application, questions of trust and reliability remain central.

We hope that this issue of EJCSRS will provide researchers, academics, practitioners, and students with useful perspectives on current developments and stimulate further investigation into the challenges identified across these rapidly developing fields. We gratefully acknowledge the contributions of the authors and the scholarly efforts involved in bringing this issue together.

Dr. Sr. Mini T V
Chief editor

CONTENTS

SL. NO	TITLE	AUTHOR	PAGE NO
1	Adversarial Attacks and Certified Defences for Deep Image Classifiers	Rasmi M	1-6
2	Diffusion Models for Image Restoration and Super Resolution: Methods and Evaluation	Rejina P V	7-11
3	Graph Neural Networks for Malware and Financial Fraud Detection	T Ramaprabha	12-16
4	Model Extraction and Membership Inference Attacks on Deployed Models	Juby George	17-21
5	Zero-Knowledge Proofs in Distributed Systems: zk-Snarks and zk-Starks For Privacy-Preserving and Scalable Blockchain Applications	Tintu George	22-27

Adversarial Attacks and Certified Defences for Deep Image Classifiers

Rasmi M

Assistant Professor, Department of Computer Applications, Bethania Institute of Management Studies,
Kunnamkulam, India

Article information

Received: 20th May 2026

Received in revised form: 23rd June 2026

Accepted: 25th July 2026

Available online: 12th September 2026

Volume: 1

Issue: 3

DOI: <https://doi.org/10.5281/zenodo.22701174>

Abstract

A change of a few pixel values, too small for a person to notice, can flip the label a deep classifier assigns to an image. Twelve years after that observation, the defensive picture has settled into two camps that are often confused with one another. Empirical defences resist the attacks that were tried against them and carry no guarantee. Certified defences return a proof that no perturbation inside a stated ball can change the prediction, and pay for that proof in accuracy, in compute, or in both. This review covers thirty two works. We describe the attack surface by what the adversary is assumed to know, we trace why a long run of empirical defences was broken by adaptive attacks, and we compare three certification strategies: exact solvers, convex relaxation with bound propagation, and randomized smoothing. Two findings deserve emphasis. Robust accuracy on standard benchmarks remains far below clean accuracy, and the gap is not closing quickly. And an evaluation that does not include an adaptive attack tells the reader almost nothing.

Keywords:- Adversarial Examples, Certified Robustness, Randomized Smoothing, Adversarial Training, Neural Network Verification, Image Classification, Threat Model.

I. INTRODUCTION

Szegedy and colleagues reported something odd in 2014. Take an image a network classifies correctly, add a perturbation chosen by optimisation, and the network changes its answer while a human sees no difference [1]. The perturbation was tiny. The failure was total. That paper started a line of work that is now large enough to need maps rather than summaries.

Goodfellow and colleagues gave the first cheap attack and a first explanation, arguing that the vulnerability comes from local linearity rather than from overfitting [2]. Papernot and colleagues mapped the attack surface [3]. Carlini and Wagner built an optimisation attack strong enough to break defences that had claimed success against weaker ones [4]. Biggio and Roli put the whole thing in the context of a decade of earlier work on adversarial machine learning [14]. Akhtar and Mian catalogued the computer vision case in particular, and the size of their bibliography by 2018 says something about how fast the area grew [15].

The reason to care is not academic. Classifiers now sit inside inspection systems, identity checks and intrusion detection pipelines [31]. An attacker who can move a decision with an imperceptible edit has a capability that ordinary software security does not cover. This review sorts what is known into attacks, empirical defences, and certification, and it states plainly which claims in each category survive scrutiny.

II. THREAT MODELS

Nothing in this field means anything without a threat model. Three things need stating: what the adversary knows, what the adversary can change, and what counts as success. Knowledge runs from full white box access through score based queries to label only queries. Figure 1 lays out the three levels with the attacks that belong to each.

The change is usually bounded by a norm. An L-infinity budget caps how far any single pixel may move, and it dominates the literature because it is easy to optimise against. L-2 budgets cap total energy. Neither corresponds well to human perception, which is a known weakness of the whole formulation. Physical attacks abandon norms altogether: Kurakin and colleagues showed that printed adversarial images survive a camera [13], and Eykholt and colleagues put stickers on road signs and moved a classifier reliably across viewing angles [12].

III. ATTACKS

White box attacks follow the gradient. FGSM takes one signed step [2]. Projected gradient descent iterates that step inside the norm ball and remains the standard baseline [7]. The Carlini and Wagner formulation replaces the constraint with a penalty and solves for the smallest perturbation that changes the label [4]. AutoAttack bundles four diverse attacks, removes the step size choices that let weak evaluations look strong, and has become the default reporting tool [25].

Black box attacks need more work but assume less. The transfer route trains a surrogate model on queried labels, attacks the surrogate, and sends the result to the target [5]. It works often enough to be a practical concern for any deployed image API. Table 1 lists the main attacks with their assumptions.

Table 1. Attacks discussed in this review, with the access each one assumes and the idea it rests on. Descriptions follow the cited papers.

Attack	Access	Core idea	Source
FGSM	White box	One signed gradient step	[2]
PGD	White box	Iterated steps, projected to the ball	[7]
Carlini and Wagner	White box	Minimise perturbation size directly	[4]
AutoAttack	White box	Ensemble with no free parameters	[25]
Surrogate transfer	Black box, labels	Attack a locally trained substitute	[5]
Physical patch	Black box	Printed marks robust to viewpoint	[12], [13]

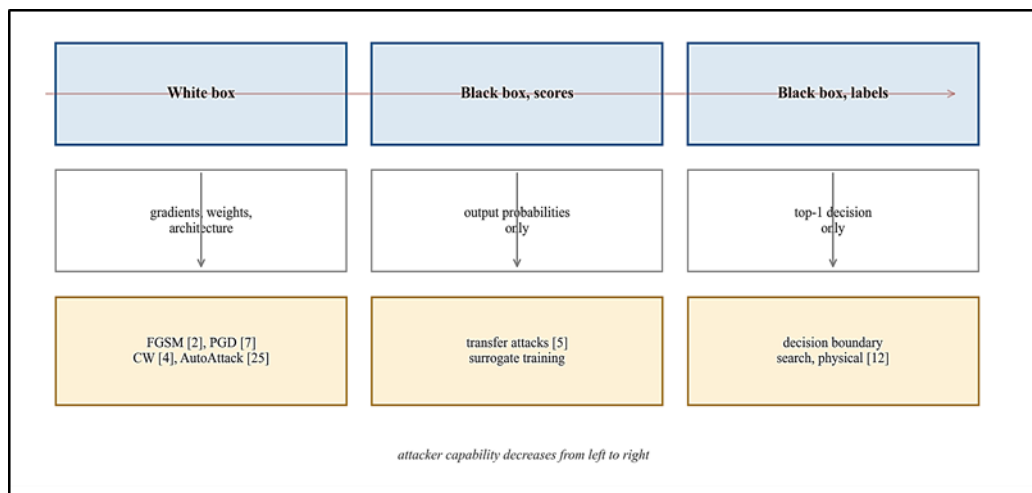


Figure. 1: Threat models ordered by adversary knowledge, with the attacks that operate at each level.

IV. EMPIRICAL DEFENCES, AND WHY SO MANY FAILED

Between 2016 and 2018 dozens of defences were published and then broken, usually within months. Athalye and colleagues explained the pattern in one paper: most of those defences did not remove the adversarial examples, they made the gradient useless to the attacker [8]. Shattered gradients, stochastic gradients and exploding gradients all produce the same symptom, which is that a gradient based attack fails while the vulnerability remains. Replace the gradient with an estimate and the defence collapses.

Tramer and colleagues extended the lesson with a set of case studies, showing that an attack adapted to the specific defence beats a standard attack nearly every time [26]. Detection based schemes such as feature squeezing [11] fall to the same treatment. The practical rule that came out of this period is simple. A defence evaluated only against fixed attacks has not been evaluated.

Adversarial training survived. Madry and colleagues framed training as a saddle point problem and solved the inner maximisation with projected gradient descent [7]. TRADES makes the trade off between clean and robust accuracy explicit in the loss [21]. Neither gives a guarantee, but neither has been broken in the way the gradient masking defences were. Rice and colleagues then found that robust overfitting is severe in this setting, and that early stopping recovers much of what elaborate schemes claim to add [27].

V. CERTIFIED DEFENCES

A. Exact verification

Exact methods answer the question completely. Reluplex extends a simplex solver to handle ReLU constraints [6], and mixed integer formulations do the same through standard optimisation machinery [23]. The answer is sound and complete. The cost grows badly with network size, so this family stays useful for small safety critical networks rather than for image classifiers of realistic depth.

B. Relaxation and bound propagation

Relaxation trades completeness for speed. Wong and Kolter bound the output over a convex outer approximation of the reachable set [9]; Raghunathan and colleagues use a semidefinite relaxation [10]; CROWN propagates linear bounds through general activations [16]; DeepPoly gives an abstract domain for the same purpose [17]. Interval bound propagation is looser still but cheap enough to include inside training, which is how Gowal and colleagues reached certified accuracy on larger images [24]. Beta-CROWN combines bound propagation with branch and bound and closes much of the gap back towards completeness [29].

C. Randomized smoothing

Smoothing takes a different path. Build a new classifier that returns the most likely label under Gaussian noise added to the input, and the resulting function is provably robust in an L_2 ball whose radius follows from the margin between the top two class probabilities [18]. The proof does not depend on the architecture, which is why this family scales to networks that verification cannot touch. Lecuyer and colleagues reached a related result through a differential privacy argument [19], and Salman and colleagues improved the radius by training the base classifier adversarially under noise [20].

The limitations are equally clear. Certification is probabilistic, since the class probabilities are estimated by sampling. Prediction becomes expensive, because each answer needs many noisy forward passes. And the guarantee is L_2 shaped, which does not transfer to the L_∞ threat model that most attacks use. Li and colleagues survey the whole certification landscape and reach similar conclusions [30].

Table 2. Certification strategies compared on the guarantee returned and the reported limits, as described by the cited works.

Strategy	Guarantee	Main limit	Source
Exact solvers	Sound and complete	Network size	[6], [23]
Convex relaxation	Sound, incomplete	Bounds loosen with depth	[9], [10], [16], [17]
Bound propagation in training	Sound, incomplete	Clean accuracy cost	[24], [29]
Randomized smoothing	Probabilistic, L_2 ball	Many forward passes per prediction	[18], [19], [20]

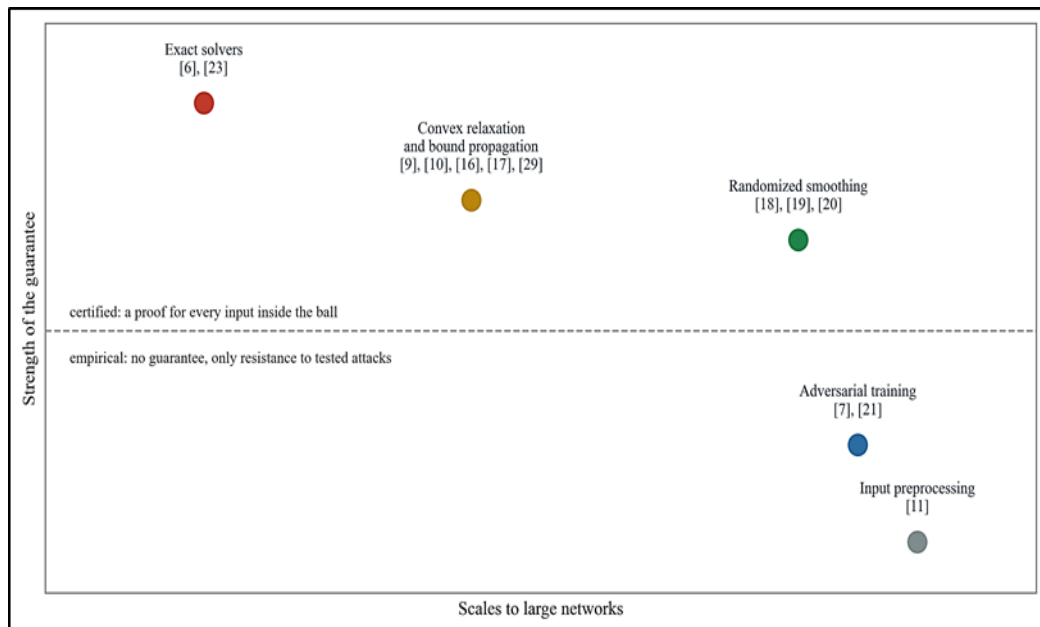


Figure. 2: Defence families placed by guarantee strength and by how far they scale. Positions are qualitative and follow the claims made in the cited papers.

VI. WHAT ROBUSTNESS COSTS

Robustness is not free, and the price shows up in three places. Clean accuracy drops: Tsipras and colleagues argue this is not an artefact of poor training but a property of the problem, since robust and accurate features can genuinely conflict [22]. Training cost rises, because every step of adversarial training runs an inner optimisation. And certified accuracy sits well below empirical robust accuracy, which itself sits well below clean accuracy.

RobustBench collects evaluations under a fixed protocol and makes the size of these gaps hard to ignore [28]. Anyone quoting a robustness number from a single paper should check it there first. Interpretability tools are sometimes proposed as a diagnostic for why a model breaks [32], though explanation and certification answer different questions and should not be substituted for one another.

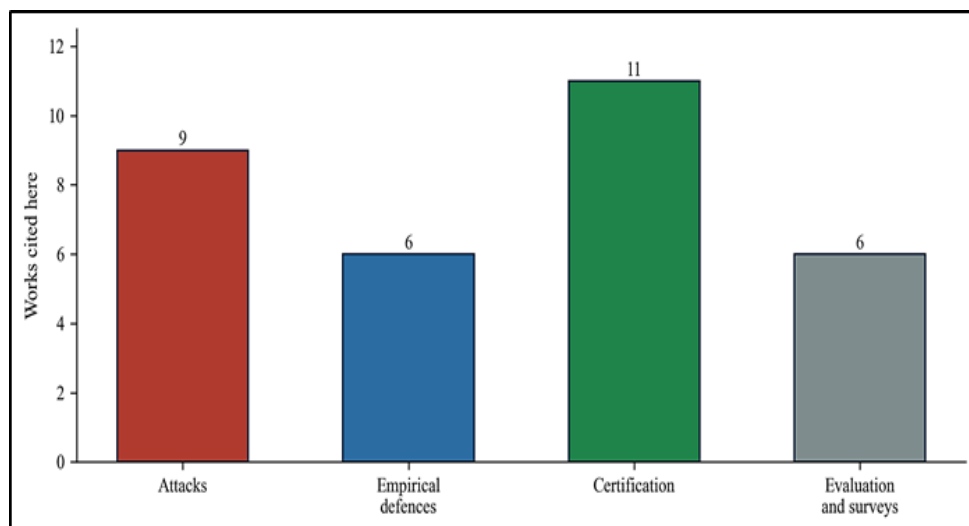


Figure. 3: The thirty two works cited here, coded by the role they play in this review. The chart counts our own reference list and reports no experimental result.

VII. OPEN PROBLEMS

Four gaps stand out. Norm bounded perturbations are a proxy for imperceptibility and a poor one, yet almost the entire literature is built on them. Certification for L-infinity threats at ImageNet scale remains out of reach. Robust overfitting has a diagnosis [27] but no clean cure. And the transfer of all of this to the physical world, where lighting, angle and printing intervene, is studied far less than the digital case [12], [13].

VIII. CONCLUSION

The field has produced one durable empirical method and one durable certified method. Adversarial training with a strong inner attack is the empirical option, and randomized smoothing is the certified option that scales. Everything else either serves smaller networks or has been broken.

For a practitioner the sequence is short. Decide the threat model before writing any code. Evaluate with AutoAttack and at least one attack adapted to your specific defence [25], [26]. Report clean accuracy, empirical robust accuracy and, where a guarantee is needed, certified accuracy with its radius. Papers that skip the third step are describing resistance, not robustness, and the difference matters when a classifier sits in a security path.

REFERENCES

- [1] C. Szegedy *et al.*, “Intriguing properties of neural networks,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014.
- [2] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.
- [3] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in *Proc. IEEE Eur. Symp. Security and Privacy (EuroS&P)*, 2016, pp. 372–387.
- [4] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *Proc. IEEE Symp. Security and Privacy (SP)*, 2017, pp. 39–57.
- [5] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in *Proc. ACM Asia Conf. Comput. Commun. Security (AsiaCCS)*, 2017, pp. 506–519.
- [6] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, “Reluplex: An efficient SMT solver for verifying deep neural networks,” in *Proc. Int. Conf. Computer Aided Verification (CAV)*, 2017, pp. 97–117.
- [7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.
- [8] A. Athalye, N. Carlini, and D. Wagner, “Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 274–283.
- [9] E. Wong and J. Z. Kolter, “Provable defenses against adversarial examples via the convex outer adversarial polytope,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 5283–5292.
- [10] A. Raghunathan, J. Steinhardt, and P. Liang, “Certified defenses against adversarial examples,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.
- [11] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” in *Proc. Netw. Distrib. Syst. Security Symp. (NDSS)*, 2018.
- [12] K. Eykholt *et al.*, “Robust physical-world attacks on deep learning visual classification,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 1625–1634.
- [13] A. Kurakin, I. J. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in *Artificial Intelligence Safety and Security*. Boca Raton, FL: Chapman and Hall/CRC, 2018, pp. 99–112.
- [14] B. Biggio and F. Roli, “Wild patterns: Ten years after the rise of adversarial machine learning,” *Pattern Recognition*, vol. 84, pp. 317–331, 2018.
- [15] N. Akhtar and A. Mian, “Threat of adversarial attacks on deep learning in computer vision: A survey,” *IEEE Access*, vol. 6, pp. 14410–14430, 2018.
- [16] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, “Efficient neural network robustness certification with general activation functions,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 4944–4953.
- [17] G. Singh, T. Gehr, M. Puschel, and M. Vechev, “An abstract domain for certifying neural networks,” *Proc. ACM Program. Lang.*, vol. 3, no. POPL, pp. 41:1–41:30, 2019.
- [18] J. Cohen, E. Rosenfeld, and J. Z. Kolter, “Certified adversarial robustness via randomized smoothing,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 1310–1320.
- [19] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, “Certified robustness to adversarial examples with differential privacy,” in *Proc. IEEE Symp. Security and Privacy (SP)*, 2019, pp. 656–672.
- [20] H. Salman, J. Li, I. Razenshteyn, P. Zhang, H. Zhang, S. Bubeck, and G. Yang, “Provably robust deep learning via adversarially trained smoothed classifiers,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 11289–11300.
- [21] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. El Ghaoui, and M. I. Jordan, “Theoretically principled trade-off between robustness and accuracy,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 7472–7482.
- [22] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, “Robustness may be at odds with accuracy,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.
- [23] V. Tjeng, K. Y. Xiao, and R. Tedrake, “Evaluating robustness of neural networks with mixed integer programming,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.
- [24] S. Gowal *et al.*, “Scalable verified training for provably robust image classification,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 4841–4850.
- [25] F. Croce and M. Hein, “Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 2206–2216.
- [26] F. Tramer, N. Carlini, W. Brendel, and A. Madry, “On adaptive attacks to adversarial example defenses,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020.
- [27] L. Rice, E. Wong, and J. Z. Kolter, “Overfitting in adversarially robust deep learning,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 8093–8104.

- [28] F. Croce *et al.*, “RobustBench: A standardized adversarial robustness benchmark,” in *Proc. NeurIPS Datasets and Benchmarks Track*, 2021.
- [29] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C.-J. Hsieh, and J. Z. Kolter, “Beta-CROWN: Efficient bound propagation with per-neuron split constraints for neural network robustness verification,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021, pp. 29909–29921.
- [30] L. Li, T. Xie, and B. Li, “SoK: Certified robustness for deep neural networks,” in *Proc. IEEE Symp. Security and Privacy (SP)*, 2023, pp. 1289–1310.
- [31] R. N, “AI-driven network security: Using deep learning to detect and mitigate cyber threats in real time,” *Int. J. Inf. Technol. Res. Stud. (IJITRS)*, vol. 1, no. 3, p. 140, Oct. 2025, doi: 10.63090/ijitrs/3139.3209.0012.
- [32] S. E, “Explainable AI (XAI): Enhancing interpretability of deep learning models for critical applications,” *Int. J. Inf. Technol. Res. Stud. (IJITRS)*, vol. 1, no. 2, p. 90, Jul. 2025, doi: 10.63090/ijitrs/3139.3209.0007.

Diffusion Models for Image Restoration and Super Resolution: Methods and Evaluation

Rejina P V

Assistant professor, Co-Operative Arts And Science College, Madayi , Pazhayangadi, Kannur, India

Article information

Received: 22nd May 2026

Received in revised form: 24th June 2026

Accepted: 27th July 2026

Available online: 12th September 2026

Volume: 1

Issue: 3

DOI: <https://doi.org/10.5281/zenodo.22701392>

Abstract

Image restoration asks for the clean picture behind a blurred, noisy, downsampled or partly missing one. That question has no single answer, and the honest output is a distribution rather than a point estimate. Diffusion models produce exactly that, which explains why they moved into restoration so quickly. This review covers thirty three works and separates the field into two routes. One trains a conditional model on paired data. The other reuses an unconditional prior and enforces agreement with the measurement at every sampling step, which needs no retraining but does need a forward model. We describe how each route handles blur, noise, downsampling and missing pixels, and we set out the sampling cost that both share. On evaluation we take a firm position: distortion metrics and perceptual metrics disagree by construction, so a paper reporting only one of them has not reported a result. We end with four open problems, of which blind degradation and hallucinated detail are the ones that matter most for scientific and medical use.

Keywords:- Diffusion Model, Image Restoration, Super Resolution, Inverse Problem, Score Matching, Perceptual Quality, Sampling Cost.

I. INTRODUCTION

Restoration problems are underdetermined. Downsample an image by four and each output pixel stands for sixteen inputs, so many clean images explain the same low resolution observation equally well. Early deep learning treated this as regression. SRCNN mapped low resolution to high resolution with three convolutional layers and a squared error loss [3]. The reconstructions scored well and looked soft, because a squared error loss returns the average of all plausible answers and the average of many sharp images is blurry.

Adversarial training was the first serious answer to that. SRGAN traded some fidelity for texture that looked real [4], ESRGAN refined the recipe [6], and Real-ESRGAN pushed it towards degradations found outside the laboratory [19]. Attention based backbones such as RCAN [7] and SwinIR [18] improved the regression side in parallel. Adversarial models are hard to train, though, and mode collapse is a real risk [13].

Diffusion models arrived from a different direction. Sohl-Dickstein and colleagues framed generation as the reversal of a gradual noising process [2]; Song and Ermon learned the score of the data distribution instead [11]; Ho and colleagues simplified the training objective into noise prediction and made the approach practical [12]. Within two years these models were beating adversarial ones on image synthesis [17]. Restoration followed. This review is about what happened next, what works, and what the reported numbers do and do not support.

II. HOW A DIFFUSION MODEL WORKS

The forward process is fixed, not learned. It takes a clean image and adds Gaussian noise in small steps until nothing recognisable is left. Because each step is Gaussian, the image at any step can be sampled directly from the clean image without walking the chain. The reverse process is the learned part: a network sees a noisy image and its noise level, and predicts the noise that was added. Subtract, repeat, and a sample emerges. Figure 1 shows the two directions.

Several refinements matter in practice. Nichol and Dhariwal learned the reverse variance and changed the noise schedule, which improved likelihood without hurting sample quality [16]. Song and colleagues showed that the discrete chain is one instance of a stochastic differential equation, and that the same trained model supports a deterministic probability flow sampler [15]. Karras and colleagues separated the design choices that papers had been bundling together, then tuned each one [21]. Rombach and colleagues moved the whole process into the latent space of an autoencoder, which cut the cost of high resolution work by a large factor [20].

III. TWO ROUTES TO RESTORATION

A. Train a conditional model

The direct approach conditions the reverse process on the degraded image. SR3 does this for super resolution: the low resolution input is concatenated to the noisy state at every step, and the model learns iterative refinement towards a plausible high resolution result [26]. Palette generalises the idea to colourisation, inpainting, uncropping and JPEG artefact removal, using one architecture and one loss across the four tasks [22].

This route is simple and it works. The cost is paired training data for each degradation you care about, plus a retraining run whenever the degradation changes. A model trained on bicubic downsampling will not handle sensor noise well.

B. Reuse an unconditional prior

The second route keeps a single unconditional diffusion model and injects the measurement during sampling. RePaint was an early example: for inpainting, replace the known region with a correctly noised version of the observation at each step, and resample to fix the seams [24]. DDRM formalises the idea for linear degradations by working in the singular value basis of the known degradation operator [23]. DDNM decomposes the solution into a range space fixed by the measurement and a null space filled in by the prior, which gives zero shot restoration across several tasks from one pretrained model [28]. Zero shot behaviour of this kind is attractive for the same reasons it is attractive in classification [32].

DPS extends the family to noisy and nonlinear forward models by approximating the gradient of the log likelihood at each step [27]. Medical imaging has taken up the approach quickly, since the forward operator in accelerated MRI is known exactly and the reconstruction is the unknown [25]. Table 1 sets the two routes side by side.

Table 1. The two restoration routes compared on requirements and coverage, following the descriptions given in the cited papers.

Property	Route A: conditional training	Route B: prior plus consistency
Training data	Paired clean and degraded	Clean images only
Retraining per task	Yes	No
Forward operator	Learned implicitly	Must be known or estimated
Degradations covered	Those seen in training	Linear [23], [28]; nonlinear [27]
Representative works	SR3 [26], Palette [22]	RePaint [24], DDRM [23], DDNM [28], DPS [27]

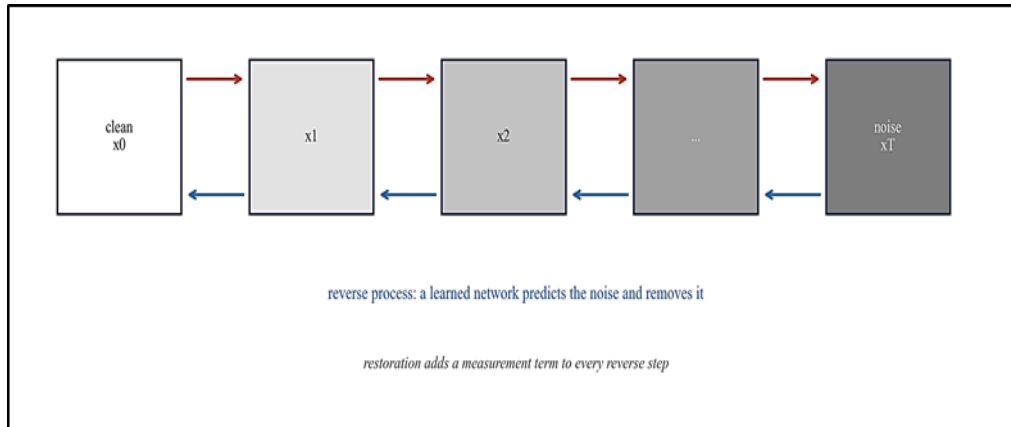


Figure. 1: The forward chain adds noise on a fixed schedule; the reverse chain removes it with a learned network. Restoration methods add a measurement term to the reverse step [23], [27], [28].

IV. SAMPLING COST

Diffusion sampling is slow. A thousand network evaluations per image was normal in the early work, against a single forward pass for a regression model. Three lines of work attack this. DDIM reformulates sampling as a non-Markovian process whose deterministic limit produces good images in tens of steps rather than hundreds [14]. Higher order and better tuned solvers for the probability flow equation cut the count further [21]. Consistency models go further still by training a network to map any point on the trajectory directly to its origin, which supports single step generation [29].

Cost matters more for restoration than for synthesis. A generative art tool can wait ten seconds. A radiology console cannot, and neither can a camera pipeline. Where the model must run near the sensor, the deployment question becomes an edge computing question rather than a modelling one [33]. Latent space diffusion helps here as well, because the expensive steps happen at reduced resolution [20].

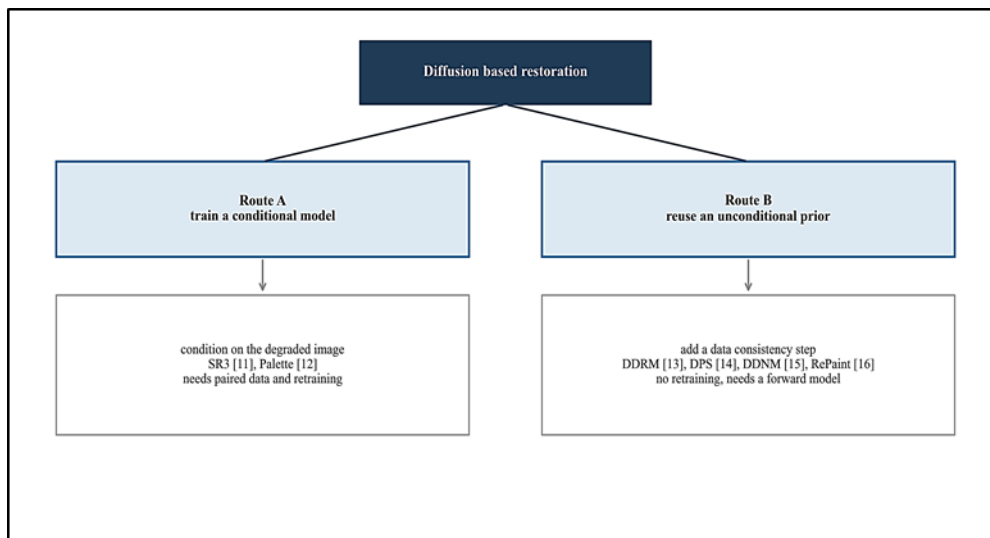


Figure. 2: The two routes used to organise this review, with the primary works assigned to each.

V. MEASURING RESTORATION QUALITY

Peak signal to noise ratio and SSIM [1] measure distortion: how close the output is to the reference. LPIPS [9] and FID [5] measure perceptual quality: whether the output looks like a real image. These two families are not merely different. Blau and Michaeli proved that they are in tension, and that improving perceptual quality past a point must cost distortion [10]. A diffusion sample sits on the perceptual side of that curve by design.

The practical consequence is blunt. A paper that reports PSNR alone will make diffusion look worse than regression. A paper that reports FID alone will make it look better than it is. Both numbers, on the same split, are the minimum. The NTIRE challenge protocol is one of the few settings where splits and degradations are pinned down well enough for cross paper comparison [8]. Table 2 lists what each metric actually measures and where it misleads.

Table 2. Metrics used across the reviewed papers, with the property each one tracks and the situation in which it is known to mislead.

Metric	Tracks	Known failure mode	Source
PSNR	Pixel level distortion	Rewards blur over texture	[10]
SSIM	Local structure and contrast	Insensitive to fine texture loss	[1]
LPIPS	Perceptual distance in feature space	Depends on the backbone network	[9]
FID	Distance between image distributions	Unstable on small sample sets	[5]

One more caution. Diffusion restoration is stochastic, so two runs give two different outputs from the same input. Reporting a single sample hides that. Where the restored image feeds a downstream measurement, the spread across samples is the number a reader needs, and very few papers report it.

VI. OPEN PROBLEMS

Blind degradation comes first. Route B assumes the forward operator is known, which is true for MRI undersampling and false for a photograph of unknown provenance. Estimating the operator and the image together remains awkward.

Hallucination comes second, and it is the one that should worry practitioners most. A diffusion model fills the null space with detail drawn from its prior. That detail is plausible and it is invented. In a photograph nobody is harmed. In a scan read for diagnosis, or in forensic imagery, an invented texture is a fabricated finding. The surveys covering this literature raise the point [30], [31], but no standard test for it has been agreed.

Third, evaluation on real degradations rather than synthetic ones. Fourth, the cost question again: single step distillation looks promising [29], yet its behaviour on restoration tasks under strict data consistency is not well characterised.

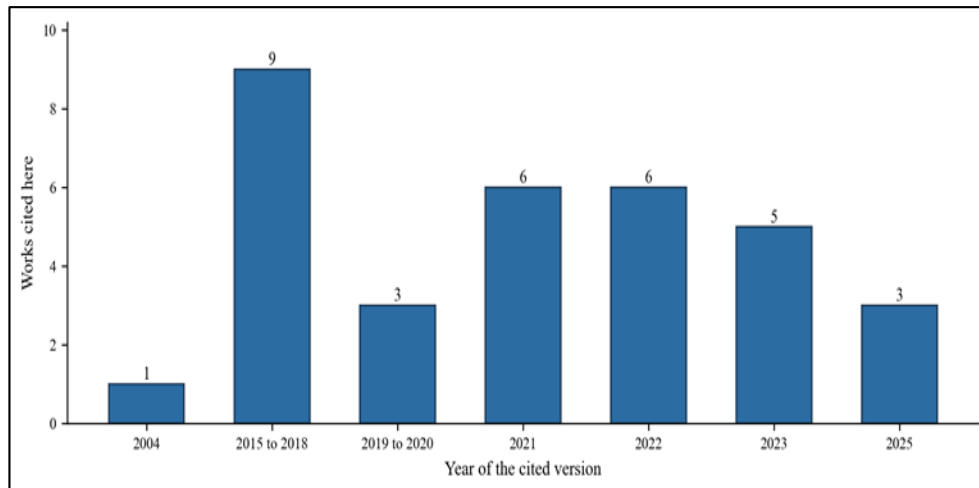


Figure. 3. Year of the cited version for the thirty three works in our reference list. The chart counts our own bibliography and reports no experimental result.

VII. CONCLUSION

Diffusion models fit restoration because restoration is a sampling problem wearing the costume of a regression problem. The plug and play route, where a pretrained unconditional prior meets a known forward operator, is the more useful of the two for most readers: one model covers many tasks, and no paired data are needed. Conditional training still wins when the degradation is fixed and paired data are plentiful.

Two recommendations follow from the reading. Report a distortion metric and a perceptual metric together, always. And treat generated detail as a hypothesis rather than a measurement, particularly in medical and scientific imaging, until a test for hallucinated structure exists.

REFERENCES

- [1] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.
- [2] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 2256–2265.
- [3] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, 2016.
- [4] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 105–114.
- [5] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 6626–6637.
- [6] X. Wang *et al.*, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. Workshops (ECCVW)*, 2018, pp. 63–79.
- [7] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 294–310.
- [8] R. Timofte, S. Gu, J. Wu, and L. Van Gool, "NTIRE 2018 challenge on single image super-resolution: Methods and results," in *Proc. CVPR Workshops*, 2018, pp. 852–863.
- [9] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. CVPR*, 2018, pp. 586–595.
- [10] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *Proc. CVPR*, 2018, pp. 6228–6237.
- [11] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 11895–11907.
- [12] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. NeurIPS*, 2020.
- [13] I. Goodfellow *et al.*, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [14] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.
- [15] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proc. ICLR*, 2021.
- [16] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proc. ICML*, 2021, pp. 8162–8171.
- [17] P. Dhariwal and A. Q. Nichol, "Diffusion models beat GANs on image synthesis," in *Proc. NeurIPS*, 2021, pp. 8780–8794.
- [18] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image restoration using Swin transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, 2021, pp. 1833–1844.
- [19] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data," in *Proc. ICCVW*, 2021, pp. 1905–1914.
- [20] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and D. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. CVPR*, 2022, pp. 10674–10685.
- [21] T. Karras, M. Aittala, A. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," in *Proc. NeurIPS*, 2022.
- [22] C. Saharia *et al.*, "Palette: Image-to-image diffusion models," in *Proc. ACM SIGGRAPH*, 2022, pp. 15:1–15:10.
- [23] B. Kawar, M. Elad, S. Ermon, and J. Song, "Denoising diffusion restoration models," in *Proc. NeurIPS*, 2022, pp. 23593–23606.
- [24] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "RePaint: Inpainting using denoising diffusion probabilistic models," in *Proc. CVPR*, 2022, pp. 11451–11461.
- [25] H. Chung and J. C. Ye, "Score-based diffusion models for accelerated MRI," *Med. Image Anal.*, vol. 80, 2022, Art. no. 102479.
- [26] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4713–4726, 2023.
- [27] H. Chung, J. Kim, M. T. McCann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," in *Proc. ICLR*, 2023.
- [28] Y. Wang, J. Yu, and J. Zhang, "Zero-shot image restoration using denoising diffusion null-space model," in *Proc. ICLR*, 2023.
- [29] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," in *Proc. ICML*, 2023, pp. 32211–32252.
- [30] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10850–10869, 2023.
- [31] X. Li *et al.*, "Diffusion models for image restoration and enhancement: A comprehensive survey," *Int. J. Comput. Vis.*, vol. 133, no. 11, pp. 8078–8108, 2025.
- [32] R. P. V., "Zero-shot learning: Enhancing models to recognize unseen classes with minimal data," *Int. J. Inf. Technol. Res. Stud. (IJITRS)*, vol. 1, no. 3, p. 157, Oct. 2025, doi: 10.63090/ijitrs/3139.3209.0014.
- [33] S. Charly, "Edge computing in networks: Reducing latency using AI-driven edge computing strategies," *Int. J. Inf. Technol. Res. Stud. (IJITRS)*, vol. 1, no. 2, p. 106, Jul. 2025, doi: 10.63090/ijitrs/3139.3209.0009.

Graph Neural Networks for Malware and Financial Fraud Detection

T Ramaprabha

Associate Professor, Department of Computer Science, Nehru Arts and Science College, Coimbatore, India

Article information

Received: 27th May 2026

Received in revised form: 30th June 2026

Accepted: 31st July 2026

Available online: 12th September 2026

Volume: 1

Issue: 3

DOI: <https://doi.org/10.5281/zenodo.22701803>

Abstract

Malware and payment fraud share a structural feature that flat feature vectors throw away. Both leave behind a graph. A binary induces a call graph; a fraud ring induces a web of accounts, devices and shared attributes. Graph neural networks learn on that structure directly, and this review examines what the move has bought in each domain. We cover thirty two works. The modelling machinery turns out to be common to both problems, while the graph construction step, which is where most of the engineering effort goes, is not. Three obstacles recur and none is solved. Fraud graphs are extremely imbalanced, so the majority class dominates the message passing. Adversaries adapt, and camouflage that connects a fraudulent node to legitimate neighbours defeats naive aggregation. And evaluation protocols that ignore time leak future information into training, which inflates reported accuracy in ways that field deployment does not reproduce. We set out what a defensible evaluation looks like and where the research gaps sit.

Keywords:- Graph Neural Network, Malware Detection, Fraud Detection, Call Graph, Message Passing, Class Imbalance, Concept Drift.

I. INTRODUCTION

Two detection problems, two industries, one shape. A malware analyst looks at a binary and asks which functions call which. A fraud analyst looks at a payment stream and asks which accounts share a device, a card, or an address. Both questions are about relations, and relations form graphs.

Traditional pipelines flattened those graphs into feature vectors. Ye and colleagues catalogued the malware side of that practice [8]; Bhattacharyya and colleagues did the same for card fraud [2]. Flattening works, up to a point. It also discards exactly the information an analyst uses when reasoning by hand, which is that this account is two hops from a known mule, or that this function calls the same encryption routine as a known dropper.

Graph neural networks remove the need to flatten. A node's representation is built from its neighbours, then from theirs, and so on for a fixed number of rounds. Kipf and Welling gave the widely used convolutional formulation [5], Hamilton and colleagues made it inductive so unseen nodes could be handled [6], Velickovic and colleagues added attention over neighbours [10], and Gilmer and colleagues showed that these are all instances of one message passing scheme [7]. This review asks how well that machinery serves two adversarial detection problems, and what breaks when it is applied carelessly.

II. FROM GRAPH TO PREDICTION

Message passing has three steps and they are worth stating plainly. Each node gathers vectors from its neighbours. It aggregates them with a permutation invariant function. It combines the result with its own current state. Repeat k times and every node has seen its k-hop neighbourhood. Figure 2 shows one round.

The aggregation function is where the models differ. Mean aggregation is smooth and loses count information. Sum aggregation keeps it, which is why Xu and colleagues showed sum based models to be more expressive in the sense of graph isomorphism testing [13]. Attention weights neighbours unequally [10], which matters when most neighbours are noise. Two surveys cover the design space in detail [18], [21].

Detection tasks come in two shapes. Malware detection is usually graph level: one binary, one label. Fraud detection is usually node level: one account, one label, inside a graph that never stops growing. Figure 1 shows both pipelines side by side. The modelling step is shared. The construction step is not, and it is where most practical difficulty lives.

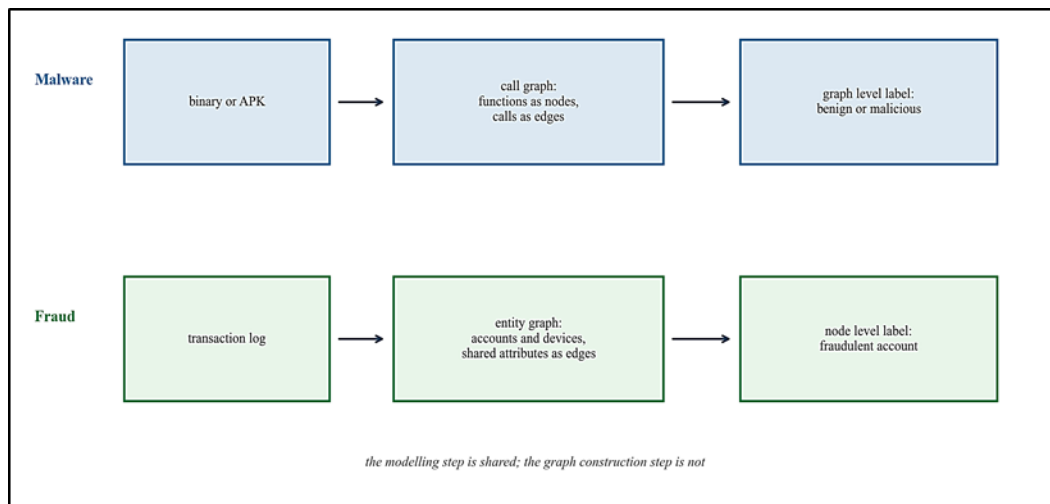


Figure. 1: Two detection pipelines. The graph construction step differs by domain; the learning step does not.

III. GRAPHS FROM BINARIES

Call graph analysis for malware predates deep learning. Hu and colleagues indexed malware by function call graph and retrieved near duplicates at scale [1]. Gascon and colleagues embedded android call graphs and classified them with a linear model [3]. DREBIN took a different route, using lightweight static features and staying explainable, and it remains a reference dataset [4]. MaMaDroid abstracted call sequences into Markov chains over packages, which gave it more resilience to small changes in the code [9].

Yan and colleagues brought graph convolution to the control flow graph directly, learning on the structure rather than on hand designed summaries of it [15]. Hei and colleagues went further and built a heterogeneous graph over apps, permissions, APIs and other entity types, then applied attention across relation types [29]. Heterogeneity matters here: an edge meaning calls and an edge meaning shares a certificate carry different evidence, and the model should not average them [14].

Two dataset problems shadow this work. Labels age, because malware families evolve, and the MalNet collection was assembled partly to give the field a large public graph resource with family labels [26]. Evaluation protocols also age badly, which is the subject of Section V. Ransomware in particular has moved fast enough that detection research struggles to keep pace with the operational picture [31].

IV. GRAPHS FROM TRANSACTIONS

Fraud graphs are built from entities and the attributes they share. Liu and colleagues modelled accounts, devices and IP addresses as a heterogeneous graph and caught malicious registrations that looked innocent one at a time [11]. Wang and colleagues added attention in a semi supervised setting, which suits fraud because labels arrive late and in small numbers [16].

Then the adversary adapted. Fraudsters connect their accounts to legitimate ones on purpose, so a fraudulent node sits in a mostly benign neighbourhood and averaging over neighbours washes out the signal. Dou and colleagues named this camouflage and answered it with a neighbour selector trained by reinforcement learning [19]. Liu and colleagues attacked the same inconsistency from the feature and relation sides [20]. Later work

combined neighbour selection with resampling to handle the extreme imbalance that fraud labels always show [22], and FRAUDRE addressed inconsistency and imbalance together [23].

Time is the other axis. Fraud arrives in bursts, so spatial structure alone misses the pattern, and spatial temporal attention has been used to capture both [24]. Tang and colleagues took a spectral view and argued that anomalies concentrate in the high frequency part of the graph signal, which suggests that the usual low pass filtering of graph convolution is working against the task [25]. That argument deserves more attention than it has received. Table 1 compares the fraud specific designs.

Table 1 Fraud specific designs and the difficulty each one targets, following the descriptions in the cited papers.

Method	Problem addressed	Mechanism	Source
Heterogeneous account graph	Entities of different types	Type aware aggregation	[11], [14]
Semi supervised attention	Few labels	Attention over neighbours	[16]
CARE-GNN	Camouflaged neighbours	Learned neighbour selection	[19]
PC-GNN	Class imbalance	Neighbour sampling and resampling	[22]
FRAUDRE	Inconsistency and imbalance	Both, in one training objective	[23]
Spatial temporal attention	Bursty behaviour	Attention across time and structure	[24]

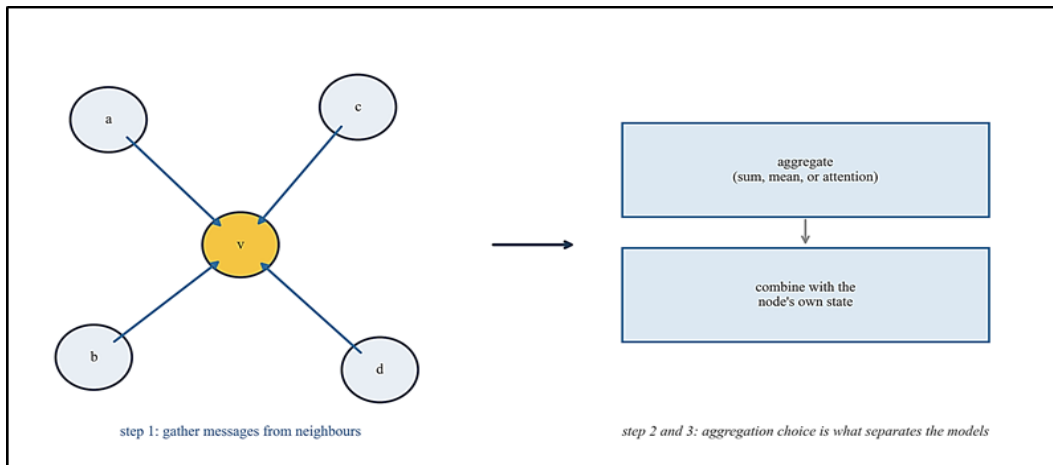


Figure. 2: One round of message passing at node v. The choice of aggregation function is what separates the models discussed in Section II [5], [6], [7], [10], [13].

V. EVALUATION, AND HOW IT GOES WRONG

This is the part of the literature that needs the most correction. Pendlebury and colleagues showed that malware classifiers evaluated with random splits report accuracies that collapse when the split respects time, because a random split lets the model learn from samples that appeared after the ones it is tested on [17]. The same criticism applies to fraud, where a random split places later transactions in training.

Three rules follow. Split by time, always. Report precision at the alert budget an analyst can actually review, since a detector that flags five percent of accounts is useless whatever its recall. And report the class balance, because accuracy on a dataset with one positive in a thousand says nothing at all. Dal Pozzolo and colleagues make the last point carefully for card fraud and add the further complication that labels arrive with a delay [12]. Table 2 lists the failure modes we found in this reading.

Table 2. Evaluation failure modes observed across the reviewed works, with the correction each one calls for.

Failure mode	Effect on reported results	Correction	Source
Random train and test split	Future information leaks backwards	Split by time	[17]
Accuracy on imbalanced data	Majority class dominates the score	Precision at a fixed alert budget	[12]
Ignoring label delay	Optimistic supervision	Simulate the real labelling lag	[12]
Single dataset reporting	Family or institution overfitting	Cross dataset evaluation	[26], [30]

Privacy is the constraint that architecture papers tend to skip. Transaction graphs contain personal financial data, and building one across institutions is a legal problem before it is a technical one. Approaches based on homomorphic encryption and differential privacy offer a path [32], but they change what the model can see, and no reviewed paper reports detection quality under such a constraint.

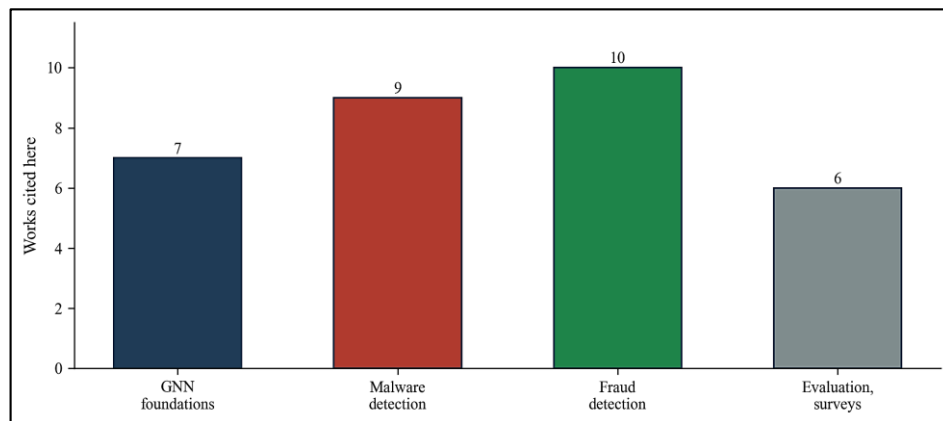


Figure. 3: The thirty two works cited here, coded by the role they play in this review. The chart counts our own reference list and reports no experimental result.

VI. OPEN PROBLEMS

Scale comes first. Payment graphs run to billions of edges and rebuild continuously, while most published evaluations use static graphs small enough to fit in memory. Second, adversarial robustness of the graph itself. An attacker who can add edges can move a prediction, and camouflage is only the simplest version of that attack.

Third, explanation. A fraud alert leads to an account freeze, and a freeze needs a reason that a human can check. Fourth, transfer across institutions and across malware families, which the surveys identify as open and which no reviewed method claims to have solved [27], [30]. Anomaly detection outside these two domains faces the same list [28].

VII. CONCLUSION

Graph neural networks earn their place in both problems, for the same reason in each: the relation between entities carries evidence that per entity features do not. The gains reported in the fraud literature are largest where the method confronts camouflage and imbalance head on rather than treating the graph as ordinary input [19], [22], [23].

Our main recommendation is about evaluation rather than architecture. A malware or fraud result from a random split should be treated as unreported until it is repeated with a temporal split [17]. Much of the apparent progress in this area would look different under that rule, and the literature would be more useful to the people who have to deploy these systems.

REFERENCES

- [1] X. Hu, T. Chiueh, and K. G. Shin, "Large-scale malware indexing using function-call graphs," in *Proc. ACM Conf. Comput. Commun. Security (CCS)*, 2009, pp. 611–620.
- [2] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.
- [3] H. Gascon, F. Yamaguchi, D. Arp, and K. Rieck, "Structural detection of android malware using embedded call graphs," in *Proc. ACM Workshop Artif. Intell. Security (AISec)*, 2013, pp. 45–54.
- [4] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, and K. Rieck, "DREBIN: Effective and explainable detection of android malware in your pocket," in *Proc. Netw. Distrib. Syst. Security Symp. (NDSS)*, 2014.
- [5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.
- [6] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1024–1034.
- [7] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1263–1272.
- [8] Y. Ye, T. Li, D. Adjeroh, and S. S. Iyengar, "A survey on malware detection using data mining techniques," *ACM Computing Surveys*, vol. 50, no. 3, pp. 41:1–41:40, 2017.
- [9] E. Mariconti, L. Onwuzurike, P. Andriotis, E. De Cristofaro, G. Ross, and G. Stringhini, "MaMaDroid: Detecting android malware by building Markov chains of behavioral models," in *Proc. NDSS*, 2017.

- [10] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. ICLR*, 2018.
- [11] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, "Heterogeneous graph neural networks for malicious account detection," in *Proc. ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2018, pp. 2077–2085.
- [12] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, 2018.
- [13] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *Proc. ICLR*, 2019.
- [14] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 2022–2032.
- [15] J. Yan, G. Yan, and D. Jin, "Classifying malware represented as control flow graphs using deep graph convolutional neural network," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, 2019, pp. 52–63.
- [16] D. Wang *et al.*, "A semi-supervised graph attentive network for financial fraud detection," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, 2019, pp. 598–607.
- [17] F. Pendlebury, F. Pierazzi, R. Jordaney, J. Kinder, and L. Cavallaro, "TESSERACT: Eliminating experimental bias in malware classification across space and time," in *Proc. USENIX Security Symp.*, 2019, pp. 729–746.
- [18] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.
- [19] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *Proc. ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2020, pp. 315–324.
- [20] Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng, "Alleviating the inconsistency problem of applying graph neural network to fraud detection," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 1569–1572.
- [21] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021.
- [22] Y. Liu *et al.*, "Pick and choose: A GNN-based imbalanced learning approach for fraud detection," in *Proc. Web Conf. (WWW)*, 2021, pp. 3168–3177.
- [23] G. Zhang *et al.*, "FRAUDRE: Fraud detection dual-resistant to graph inconsistency and imbalance," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, 2021, pp. 867–876.
- [24] D. Cheng, X. Wang, Y. Zhang, and L. Zhang, "Graph neural network for fraud detection via spatial-temporal attention," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3800–3813, 2022.
- [25] J. Tang, J. Li, Z. Gao, and J. Li, "Rethinking graph neural networks for anomaly detection," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2022, pp. 21076–21089.
- [26] S. Freitas, R. Duggal, and D. H. Chau, "MalNet: A large-scale image database of malicious software," in *Proc. ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2022, pp. 3948–3952.
- [27] G. Pang, C. Shen, L. Cao, and A. van den Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys*, vol. 54, no. 2, pp. 38:1–38:38, 2022.
- [28] Y. Wu, H.-N. Dai, and H. Tang, "Graph neural networks for anomaly detection in industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9214–9231, 2022.
- [29] Y. Hei *et al.*, "Hawk: Rapid android malware detection through heterogeneous graph attention networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4703–4717, 2024.
- [30] S. Motie and B. Raahemi, "Financial fraud detection using graph neural networks: A systematic review," *Expert Systems with Applications*, vol. 240, 2024, Art. no. 122156.
- [31] K. Abraham, "Ransomware evolution: How AI is changing both attack and defence strategies," *Int. J. Inf. Technol. Res. Stud. (IJITRS)*, vol. 1, no. 3, p. 164, Oct. 2025, doi: 10.63090/ijitrs/3139.3209.0015.
- [32] M. J. Komban, "Privacy-preserving techniques in data mining: A comprehensive analysis of homomorphic encryption and differential privacy approaches," *Int. J. Inf. Technol. Res. Stud. (IJITRS)*, vol. 1, no. 2, p. 127, Jul. 2025, doi: 10.63090/ijitrs/3139.3209.0011.

Model Extraction and Membership Inference Attacks on Deployed Models

Juby George

Assistant Professor, Department of Computer Applications, Marian College Kuttikkanam Autonomous, India

Article information

Received: 2nd June 2026

Received in revised form: 4th July 2026

Accepted: 5th August 2026

Available online: 12th September 2026

Volume: 1

Issue: 3

DOI: <https://doi.org/10.5281/zenodo.22702167>

Abstract

A prediction interface gives away more than a prediction. Query it enough times and an attacker can rebuild a working copy of the model, decide whether a particular record was in its training set, or in some settings recover the record itself. This review covers thirty three works and separates two attack families that are often treated as one. Extraction targets the model as intellectual property; inference targets the people whose data trained it. We trace the progression from the first prediction API attack to modern likelihood ratio tests that succeed at low false positive rates, and we explain why average case accuracy is the wrong way to report a membership attack. On the defensive side we group techniques into prevention, obstruction and attribution, and we note that only the first carries a formal guarantee. Two conclusions stand out. Overfitting drives membership leakage but does not fully explain it. And watermarking, the industry favourite for extraction, has weaker guarantees than its adoption would suggest.

Keywords:- Model Extraction, Membership Inference, Machine Learning Privacy, Differential Privacy, Model Watermarking, Prediction API, Threat Model.

I. INTRODUCTION

Machine learning is sold as a service. A client sends an input and receives a prediction, and the weights stay on the provider side. That arrangement is supposed to protect two things at once: the provider's investment in training, and the privacy of whoever supplied the training data. It protects neither as well as it appears to.

Tramer and colleagues showed the first half in 2016. With enough queries to a prediction API you can reconstruct a model that behaves like the original, sometimes exactly, and the query budget was small enough to be affordable [3]. Shokri and colleagues showed the second half a year later: given only the output for a record, an attacker can often tell whether that record was used in training [5]. For a model trained on medical records, that is a disclosure about a person, not about a model.

The two attacks are frequently discussed together and they have different victims. Extraction is a theft of intellectual property. Inference is a privacy breach. The defences differ accordingly, and conflating them leads to the mistake of deploying a watermark and believing the training data are now protected. Figure 1 lays out the shared attack surface. Three surveys cover parts of this ground [26], [29], [31]; our aim is to state the distinction sharply and to be direct about what the defences guarantee.

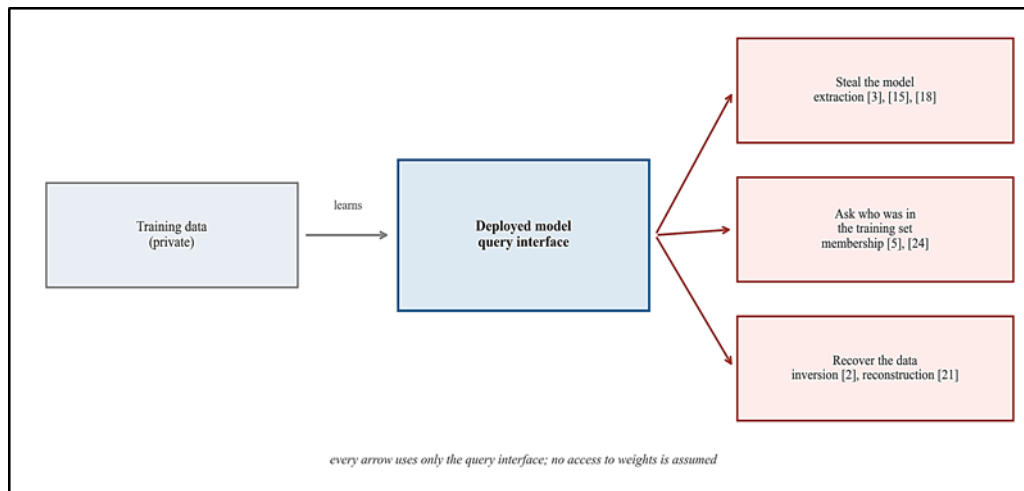


Figure. 1: One interface, three goals. Extraction targets the model, inference targets the training data, and both work through the same query channel.

II. STEALING THE MODEL

Extraction has two goals that pull in different directions. High accuracy extraction wants a model that performs well on the task. High fidelity extraction wants a model that agrees with the target on every input, including the inputs where the target is wrong. Jagielski and colleagues drew that distinction and showed the second is much harder [18].

Method matters less than query strategy. Knockoff nets query the target with images drawn from an unrelated public pool, then train a student on the returned scores, and reach useful accuracy without any data from the original distribution [15]. Chandrasekaran and colleagues made the connection to active learning explicit, which reframes extraction as a query efficiency problem with a well studied theory behind it [19]. For simple model classes, exact recovery through equation solving is possible [3].

Defences fall into three layers, shown in Figure 2. Obstruction perturbs the returned confidence scores, trading a little utility for a worse training signal [17], or watches the distribution of incoming queries for the unnatural patterns that extraction produces [14]. Attribution accepts that the model may be stolen and tries to prove it afterwards, which is what watermarking does: embed a signal in the weights [7], or in the behaviour on a secret trigger set [9], or entangle the mark with the task itself so that removing it costs accuracy [23].

The uncomfortable finding is about watermarking. Lukas and colleagues evaluated image classification watermarks against a broad set of removal attacks and found that most schemes fail under an adversary who is willing to spend a modest amount of compute [25]. Watermarking is worth deploying. It is not worth relying on.

Table 1. Extraction attacks and defences, with what each one assumes or provides, following the cited papers.

Technique	Type	Assumption or guarantee	Source
Equation solving	Attack	Simple model class, score outputs	[3]
Knockoff nets	Attack	Unrelated public query pool	[15]
Active learning query design	Attack	Query budget is the binding cost	[19]
Output perturbation	Defence	Utility loss, no guarantee	[17]
Query monitoring	Defence	Detects abnormal query patterns	[14]
Watermarking	Defence	Attribution only, removal shown feasible	[7], [9], [23], [25]

III. INFERRING THE TRAINING DATA

A. Membership inference

The question is narrow and the consequences are not. Was this record in the training set? Shokri and colleagues answered it by training shadow models on data from a similar distribution, then training a classifier to recognise the output patterns a model produces for members [5]. Salem and colleagues stripped away most of the assumptions and showed the attack still works with a single shadow model, or none [12].

Yeom and colleagues connected leakage to overfitting, which gave the field a useful intuition and, taken too far, a misleading one [8]. Overfitting is sufficient for membership leakage. It is not necessary. Well generalised models still leak on their atypical training points, which are exactly the records that identify individuals.

Choquette-Choo and colleagues removed the last convenient assumption by mounting the attack with only the predicted label, no confidence scores [20]. Nasr and colleagues extended the analysis to white box and federated settings, where an adversary who sees updates has more to work with than one who sees outputs [13]. Zhu and colleagues showed that shared gradients alone can leak the training input [16], which bears directly on federated deployments [33].

B. Reporting these attacks honestly

Carlini and colleagues made an argument that changed how this work should be reported [24]. A membership attack with fifty five percent average accuracy sounds weak. But an attack that is right about a small number of records with near certainty is a serious breach, and average accuracy hides it. They recommend reporting the true positive rate at a low fixed false positive rate, and evaluating per example rather than in aggregate. Song and Mittal reached a similar conclusion from a different direction [22], and Ye and colleagues built a hypothesis testing framework around it [28].

Anyone reading older results should apply the correction. Papers reporting only average accuracy understate the risk for the most vulnerable records, and those are the records that matter.

C. Beyond membership

Model inversion reconstructs a representative input for a class rather than deciding membership, which Fredrikson and colleagues demonstrated on face recognition [1]. Property inference recovers global facts about the training distribution that the provider never intended to publish [10]. Extraction of verbatim training data is now documented for large language models [21] and for diffusion models, where individual training images can be regenerated [30]. Tramer and colleagues added a poisoning step that raises the leakage of targeted records, which turns a passive attacker into an active one [27]. Table 2 sets out the family.

Table 2. Inference attacks grouped by what the adversary learns and what access is required, as reported in the cited works.

Attack	What is learned	Access needed	Source
Membership inference	Was this record trained on	Scores, or labels only	[5], [12], [20]
White box and federated	Membership, from updates	Model internals or gradients	[13], [16]
Model inversion	A class representative input	Scores	[1]
Property inference	Facts about the training set	Model access	[10]
Verbatim extraction	Actual training records	Generation interface	[21], [30]

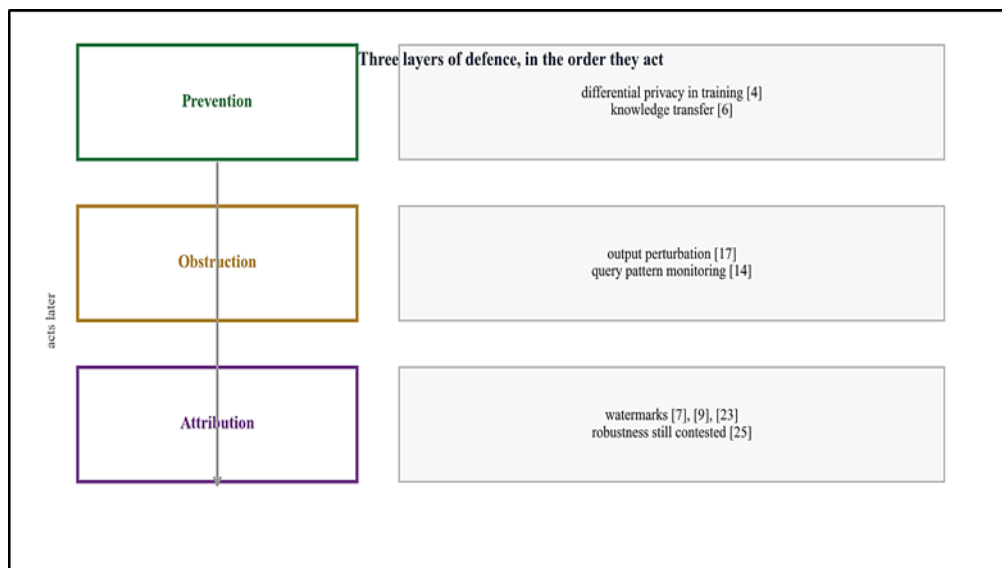


Figure. 2: Defences grouped by when they act. Only the prevention layer carries a formal guarantee.

IV. DEFENCES AND WHAT THEY GUARANTEE

Differential privacy is the only prevention with a formal guarantee. The definition bounds how much any single record can change the output distribution [2], and DP-SGD applies it to deep learning by clipping per example gradients and adding calibrated noise [4]. PATE reaches a similar end through knowledge transfer from an ensemble of teachers trained on disjoint partitions [6]. The guarantee is real and it is bought with accuracy, particularly on the rare classes that matter most in medical and financial work.

Empirical defences carry no such guarantee. Adversarial regularisation trains the model against a membership attacker and reduces leakage against the attacks tested [11]. Whether it holds against a stronger attacker is the same open question that dogs empirical defences elsewhere in security. Encryption based approaches change the picture differently by keeping data confidential during computation rather than bounding what the model memorises [32].

A note on privacy budgets. The epsilon values used in deployed systems are often far larger than the theory's comfortable range, and a large epsilon means the guarantee is weak even though it is technically present. Reporting the value, rather than reporting that differential privacy was used, is the minimum a reader needs.

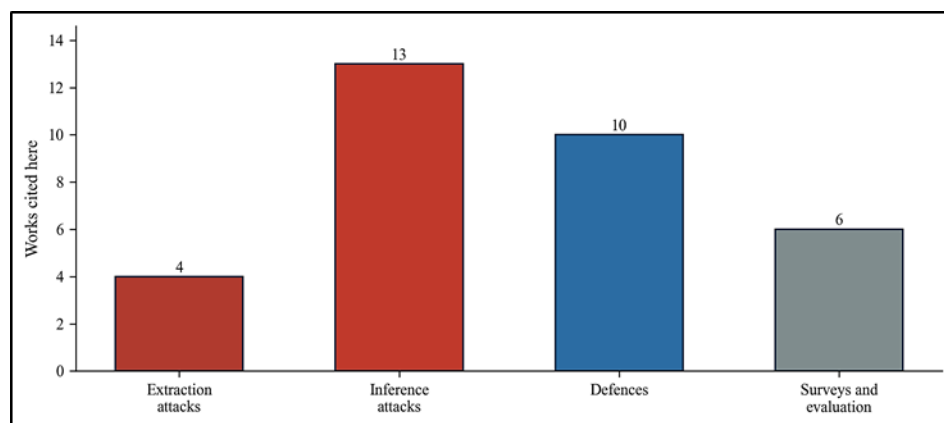


Figure. 3: The thirty three works cited here, coded by the role they play in this review. The chart counts our own reference list and reports no experimental result.

V. OPEN PROBLEMS

Auditing comes first. Given a deployed model, how much did it actually leak? Formal bounds are loose and empirical attacks are lower bounds, so the true value sits in a wide interval. Second, generative models. Extraction and memorisation behave differently there [21], [30], and the defences designed for classifiers do not transfer cleanly.

Third, the cost of privacy falls unevenly, and the classes that lose most accuracy under DP-SGD tend to be the small ones. Fourth, the legal position. A right to erasure implies removing a record's influence from a trained model, and no reviewed method achieves that without retraining.

VI. CONCLUSION

A prediction interface is an information channel, and this literature is the study of what flows through it. Extraction and inference use the same channel and threaten different parties, so a provider needs a position on both.

Three practical points close the review. Use differential privacy where the training data are personal, and report epsilon rather than the fact of its use [2], [4]. Evaluate membership risk at low false positive rates, not on average [24]. And treat watermarking as evidence for a later dispute rather than as a defence, given how it performs under removal attacks [25].

REFERENCES

- [1] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. ACM Conf. Comput. Commun. Security (CCS)*, 2015, pp. 1322–1333.
- [2] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory of Cryptography Conf. (TCC)*, 2006, pp. 265–284.

- [3] F. Tramer, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in *Proc. USENIX Security Symp.*, 2016, pp. 601–618.
- [4] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. CCS*, 2016, pp. 308–318.
- [5] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Security and Privacy (SP)*, 2017, pp. 3–18.
- [6] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.
- [7] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proc. ACM Int. Conf. Multimedia Retrieval (ICMR)*, 2017, pp. 269–277.
- [8] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *Proc. IEEE Comput. Security Found. Symp. (CSF)*, 2018, pp. 268–282.
- [9] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *Proc. USENIX Security Symp.*, 2018, pp. 1615–1631.
- [10] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, "Property inference attacks on fully connected neural networks using permutation invariant representations," in *Proc. CCS*, 2018, pp. 619–633.
- [11] M. Nasr, R. Shokri, and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *Proc. CCS*, 2018, pp. 634–646.
- [12] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *Proc. Netw. Distrib. Syst. Security Symp. (NDSS)*, 2019.
- [13] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. SP*, 2019, pp. 739–753.
- [14] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, "PRADA: Protecting against DNN model stealing attacks," in *Proc. IEEE Eur. Symp. Security and Privacy (EuroS&P)*, 2019, pp. 512–527.
- [15] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 4949–4958.
- [16] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 14747–14756.
- [17] T. Orekondy, B. Schiele, and M. Fritz, "Prediction poisoning: Towards defenses against DNN model stealing attacks," in *Proc. ICLR*, 2020.
- [18] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, "High accuracy and high fidelity extraction of neural networks," in *Proc. USENIX Security Symp.*, 2020, pp. 1345–1362.
- [19] V. Chandrasekaran, K. Chaudhuri, I. Giacomelli, S. Jha, and S. Yan, "Exploring connections between active learning and model extraction," in *Proc. USENIX Security Symp.*, 2020, pp. 1309–1326.
- [20] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 1964–1974.
- [21] N. Carlini et al., "Extracting training data from large language models," in *Proc. USENIX Security Symp.*, 2021, pp. 2633–2650.
- [22] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," in *Proc. USENIX Security Symp.*, 2021, pp. 2615–2632.
- [23] H. Jia, C. A. Choquette-Choo, V. Chandrasekaran, and N. Papernot, "Entangled watermarks as a defense against model extraction," in *Proc. USENIX Security Symp.*, 2021, pp. 1937–1954.
- [24] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *Proc. SP*, 2022, pp. 1897–1914.
- [25] N. Lukas, E. Jiang, X. Li, and F. Kerschbaum, "SoK: How robust is image classification deep neural network watermarking?," in *Proc. SP*, 2022, pp. 787–804.
- [26] H. Hu, Z. Salicic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey," *ACM Computing Surveys*, vol. 54, no. 11s, pp. 235:1–235:37, 2022.
- [27] F. Tramer, R. Shokri, A. San Joaquin, H. Le, M. Jagielski, S. Hong, and N. Carlini, "Truth serum: Poisoning machine learning models to reveal their secrets," in *Proc. CCS*, 2022, pp. 2779–2792.
- [28] J. Ye, A. Maddi, S. K. Murakonda, V. Bindschaedler, and R. Shokri, "Enhanced membership inference attacks against machine learning models," in *Proc. CCS*, 2022, pp. 3093–3106.
- [29] D. Oliynyk, R. Mayer, and A. Rauber, "I know what you trained last summer: A survey on stealing machine learning models and defences," *ACM Computing Surveys*, vol. 55, no. 14s, pp. 324:1–324:41, 2023.
- [30] N. Carlini et al., "Extracting training data from diffusion models," in *Proc. USENIX Security Symp.*, 2023, pp. 5253–5270.
- [31] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," *ACM Computing Surveys*, vol. 56, no. 4, pp. 101:1–101:34, 2024.
- [32] M. J. Komban, "Privacy-preserving techniques in data mining: A comprehensive analysis of homomorphic encryption and differential privacy approaches," *Int. J. Inf. Technol. Res. Stud. (IJITRS)*, vol. 1, no. 2, p. 127, Jul. 2025, doi: 10.63090/ijitrs/3139.3209.0011.
- [33] K. Abraham, "Federated learning: Privacy-preserving distributed training," *Eduschool J. Comput. Sci. Res. Stud. (EJCSRS)*, vol. 1, no. 1, pp. 22–26, Mar. 2026, doi: 10.5281/zenodo.18979096.

Zero-Knowledge Proofs in Distributed Systems: zk-Snarks and zk-Starks For Privacy-Preserving and Scalable Blockchain Applications

Tintu George

Assistant Professor, Department of BCA AI, Sri Ramakrishna College of Arts & Science, Coimbatore, India

Article information

Received: 8th June 2026

Received in revised form: 10th July 2026

Accepted: 11th August 2026

Available online: 12th September 2026

Volume: 1

Issue: 3

DOI: <https://doi.org/10.5281/zenodo.22702428>

Abstract

A zero-knowledge proof lets one party convince another that a statement is true while revealing nothing beyond the fact of its truth. The idea, introduced in cryptographic theory in the 1980s, sounded paradoxical and remained largely academic for thirty years. It has since become one of the most consequential tools in distributed systems, because a blockchain is exactly the setting where the paradox pays off: every participant must be convinced that a transaction is valid, yet privacy demands that the details stay hidden, and scalability demands that verifying a computation cost far less than performing it. Modern succinct proof systems deliver both. This paper explains zero-knowledge proofs for a systems audience and contrasts the two families now driving deployment. zk-SNARKs produce extremely small proofs that verify in milliseconds but have historically required a trusted setup whose compromise would be catastrophic. zk-STARKs eliminate that trusted setup and resist quantum attack, at the cost of substantially larger proofs. We cover the properties that define a zero-knowledge argument, the path from interactive proofs to non-interactive succinct ones, the concrete trade-offs between the two families, and the two applications reshaping blockchains: confidential transactions that hide amounts and parties, and validity rollups that move computation off-chain while proving its correctness on-chain.

Keywords:- Zero-Knowledge Proofs, ZK-SNARK, ZK-STARK, Blockchain, Privacy, Verifiable Computation, Succinct Arguments, Trusted Setup, Rollups, Post-Quantum Cryptography.

I. INTRODUCTION

Among the stranger results in theoretical computer science is the proof that you can convince someone a statement is true without telling them anything about why. Goldwasser, Micali, and Rackoff formalised this in 1985 as the zero-knowledge proof, a protocol satisfying three properties at once: an honest prover can always convince the verifier of a true statement, no prover can convince the verifier of a false one except with negligible probability, and the verifier learns nothing from the interaction beyond the statement's truth [1], [2]. For decades this was a beautiful idea with few practical consumers.

Blockchains changed that. A public ledger is a distributed system with two requirements in tension. Every node must independently verify that each transaction obeys the rules, which seems to demand that every node see every detail; yet users want privacy, and the network wants to avoid the cost of having every node re-execute every computation [8], [13]. Zero-knowledge proofs resolve both tensions with one mechanism. A proof can establish that a transaction is valid without revealing its amounts or participants, and a proof can establish that a large off-chain computation was performed correctly while costing the verifier almost nothing to check [4], [14].

The same primitive thus serves privacy and scalability, which is why it has moved from theory to the centre of blockchain engineering.

This paper introduces zero-knowledge proofs for systems engineers and compares the two practical families in deployment. Section II states the defining properties and Figure 1 depicts the prover-verifier relationship. Section III traces the route from interactive proofs to the succinct, non-interactive arguments that real systems use. Sections IV and V contrast zk-SNARKs and zk-STARKs and the trade-offs that distinguish them, and Sections VI and VII examine the privacy and scaling applications and the open challenges that remain.

II. WHAT A ZERO-KNOWLEDGE PROOF GUARANTEES

A zero-knowledge argument is defined by three properties, and grasping them is enough to understand why the technology is useful even without its mathematics [1]. Completeness says that if the statement is true and both parties follow the protocol, the verifier will be convinced. Soundness says that if the statement is false, no cheating prover can convince the verifier except with vanishingly small probability, so a proof that passes can be trusted. Zero-knowledge, the third and most surprising property, says that the verifier learns nothing from the proof except that the statement is true; in particular it learns nothing about the prover's secret, the witness, that makes the statement true [1], [2]. Figure 1 shows the setting: a prover holding a secret witness for a public statement produces a proof, and a verifier checks it against the public statement alone. The classic illustration is proving you know a password without sending it, but the general form, proving you correctly performed an arbitrary computation without revealing its private inputs, is what makes the primitive so powerful in distributed systems [15], [17].

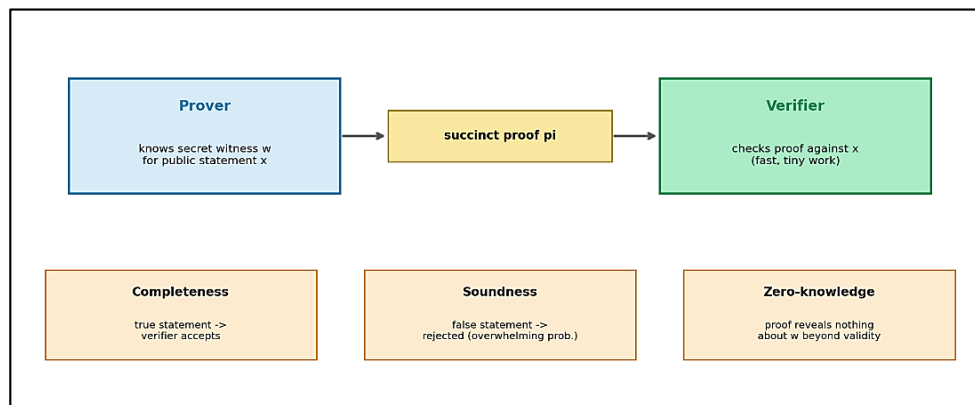


Fig 1: A zero-knowledge argument. The prover convinces the verifier that it holds a valid witness for a public statement; completeness, soundness, and the zero-knowledge property together let the verifier trust the claim while learning nothing about the secret.

III. FROM INTERACTIVE PROOFS TO SUCCINCT ARGUMENTS

A. Removing the Interaction

The original zero-knowledge proofs were interactive: prover and verifier exchanged several rounds of challenges and responses [1], [2]. That is unworkable on a blockchain, where a proof must be posted once and checked later by many parties who cannot interact with the prover. The Fiat-Shamir heuristic and the subsequent theory of non-interactive proofs solved this by letting the prover generate the verifier's challenges itself from a cryptographic hash, collapsing the dialogue into a single message that anyone can verify [5], [15]. Non-interactivity is the first property a deployable system needs.

B. Making the Proof Succinct

The second property is succinctness: the proof must be small and fast to verify, ideally far smaller than the computation it attests to, so that verifying is much cheaper than re-executing. This is the property captured by the term SNARK, a succinct non-interactive argument of knowledge [3]. A line of work running from Pinocchio [9] through quadratic span programs [5] to the highly efficient Groth16 construction [6] made such arguments practical, reducing a proof of an arbitrary computation to a handful of group elements that a verifier checks with a constant, tiny amount of work regardless of how large the original computation was [17]. Verifiable computation in this sense, proving that a program ran correctly on hidden inputs, is the engine beneath every application in this paper [4], [16].

IV. zk-SNARKs

zk-SNARKs are the family that brought zero-knowledge proofs into production, and their defining strength is extreme succinctness. A Groth16-class proof is only a few hundred bytes and verifies in milliseconds, independent of the size of the computation proved, which is precisely what a blockchain needs when thousands of nodes must each check the proof cheaply [4], [6]. This efficiency is what made the first privacy-preserving cryptocurrency, which hides transaction amounts and participants behind zk-SNARK proofs of validity, technically feasible [8].

The strength comes with a serious caveat: most efficient zk-SNARKs require a trusted setup. Before the system can be used, a set of public parameters must be generated from secret randomness, and if anyone learns that randomness, often called the toxic waste, they can forge proofs of false statements undetectably [4], [11]. The security of the entire system thus rests on that secret having been destroyed. The practical mitigation is a multi-party computation ceremony in which many participants jointly generate the parameters such that the setup is secure as long as even one participant was honest and discarded their share [11]. Newer constructions reduce the burden further by making the setup universal and updatable, so a single ceremony serves many programs rather than requiring a fresh one per circuit [12]. Still, the trusted setup remains the principal structural weakness of the SNARK approach.

V. zk-STARKs AND THE TRADE-OFF

zk-STARKs were designed to remove exactly that weakness. They are transparent, requiring no trusted setup at all, because they rely only on cryptographic hash functions and public randomness rather than secret parameters, which eliminates the toxic-waste risk entirely [7]. A second consequence of building on hash functions is post-quantum security: unlike pairing-based zk-SNARKs, whose hardness assumptions a large quantum computer would break, STARKs rest on assumptions believed to survive quantum attack [7]. They also scale well for very large computations, with prover overhead growing only quasi-linearly in the size of the statement. The cost is proof size. A STARK proof is typically tens to hundreds of kilobytes, orders of magnitude larger than a SNARK proof, and verification, while still far cheaper than re-execution, is correspondingly heavier [7]. On a blockchain, where every byte stored and every verification performed has a direct cost, this is a real disadvantage. The comparison therefore has no universal winner, and the choice is an engineering trade-off, illustrated in Figure 2 and summarised in Table 1: SNARKs minimise on-chain footprint at the price of a trusted setup and quantum vulnerability, while STARKs accept larger proofs in exchange for transparency and post-quantum security [6], [7], [12].

Table 1. zk-SNARKs Versus zk-STARKs

Property	zk-SNARK (Groth16-class)	zk-STARK
Proof size	Very small (hundreds of bytes)	Large (tens-hundreds of KB)
Verification cost	Very low (ms, near-constant)	Higher but still sublinear
Trusted setup	Required (per-circuit or universal)	None (transparent)
Post-quantum secure	No (pairing-based assumptions)	Yes (hash-based)
Cryptographic basis	Elliptic-curve pairings	Collision-resistant hashes

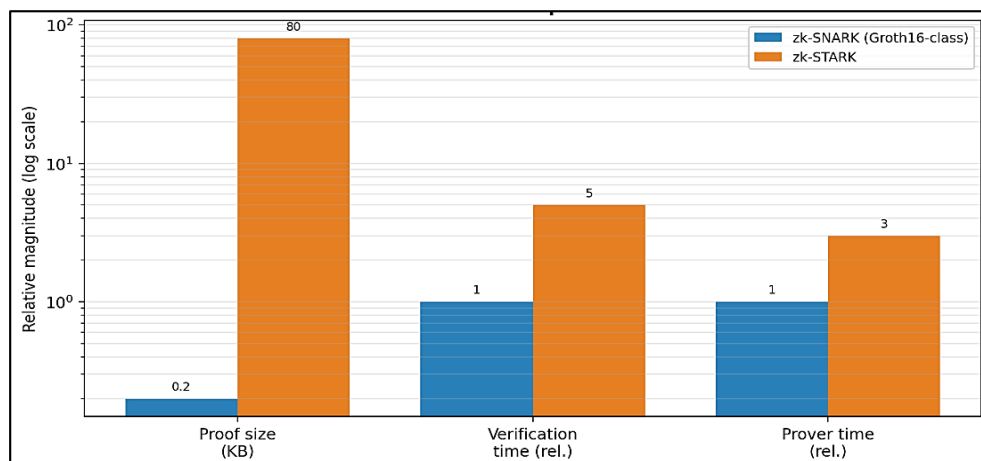


Fig 2: Representative trade-offs between the two families on a log scale. zk-SNARKs win decisively on proof size and verification cost; zk-STARKs pay for that with much larger proofs but need no trusted setup and resist quantum attack.

VI. APPLICATIONS IN DISTRIBUTED LEDGERS

A. Confidential Transactions

The first application is privacy. On a transparent ledger every transaction is public, which is unacceptable for many real uses. Zero-knowledge proofs let a user prove that a transaction is valid, that the inputs exist, that they are authorised to spend, and that inputs balance outputs, without revealing the amounts or the parties involved [8]. The pioneering deployment used zk-SNARKs to realise fully shielded payments on a public blockchain, and related techniques such as Bulletproofs provide compact range proofs that hide transaction amounts without any trusted setup [8], [10]. The general pattern is to replace public data with a proof that the hidden data satisfies the rules.

B. Validity Rollups for Scaling

The second application, and the one currently driving the most engineering investment, is scaling. A blockchain's throughput is limited because every node re-executes every transaction. A validity rollup, often called a zk-rollup, breaks this limit by executing a large batch of transactions off-chain and posting to the main chain only a succinct proof that the entire batch was processed correctly, together with the resulting state [14]. Every node then verifies one small proof instead of re-executing thousands of transactions, so the chain inherits the security of full verification at a fraction of the cost [4], [14]. This is verifiable computation applied at scale, and it is the leading approach to scaling smart contract platforms without weakening their security guarantees [13], [14]. Tooling that compiles high-level programs into provable circuits has made building such systems accessible to ordinary developers rather than only cryptographers [16].

VII. INSIDE A SNARK: FROM PROGRAMS TO POLYNOMIALS

It is worth opening the box, because the internal pipeline explains both the power and the fragility of these systems. A proof system cannot reason about a program directly; the computation must first be expressed as an arithmetic circuit, a network of addition and multiplication gates over a finite field that computes the same result [9], [17]. The circuit is then written as a constraint system, most commonly a rank-one constraint system, in which each gate becomes an equation that a correct execution must satisfy, so that proving the computation was done correctly reduces to proving that a set of assignments satisfies all the constraints [5], [9]. The next step is the conceptual heart of succinctness: the satisfaction of all those constraints is encoded into polynomials, and the prover commits to those polynomials so that the verifier can check a few random points instead of every constraint, relying on the fact that distinct low-degree polynomials almost never agree at a random point [5], [20]. Figure 3 traces this pipeline from program to proof. The polynomial-commitment scheme is the component that most distinguishes modern systems from one another, and advances in commitments have driven much of the recent progress [20]. The same pipeline also explains the dominant security risk noted earlier: a mistake in translating a computation into a circuit produces a proof that is perfectly valid for the wrong statement, which is why circuit correctness and auditing matter as much as the cryptography [16].

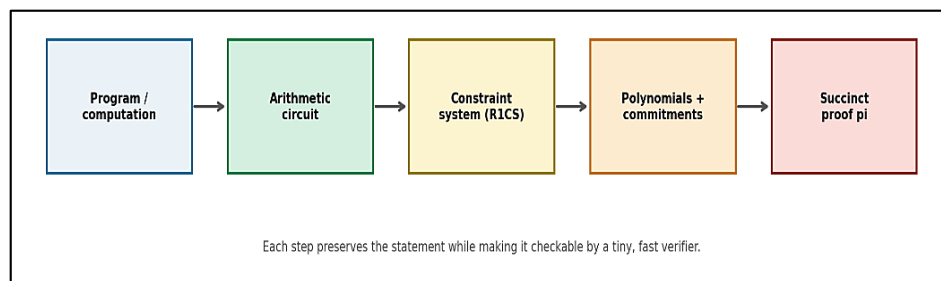


Fig 3: The internal pipeline of a succinct proof. A computation becomes an arithmetic circuit, then a constraint system, then polynomials the prover commits to, letting the verifier check a few random points rather than the whole computation.

VIII. THE MODERN PROOF-SYSTEM LANDSCAPE

The clean two-family split of Sections IV and V is a useful simplification of a landscape that has grown rich. The pairing-based construction that minimises proof size also fixes its parameters to a single circuit, so a change to the program demands a fresh trusted setup [6]. A major line of work removed that rigidity with universal and updatable setups, in which one ceremony produces parameters reusable across many programs, balancing succinctness against operational practicality [12], [21]. A separate family abandons the trusted setup entirely while keeping proofs far smaller than STARKs: inner-product arguments yield short proofs for confidential transactions and range statements with no setup at all, at the cost of verification that grows with the statement [10], [22]. Transparent constructions have also been built from polynomial commitments that need no secret parameters,

narrowing the gap between the SNARK and STARK worlds from the SNARK side [19]. Meanwhile the hash-based, FRI-driven STARK approach continues to anchor the transparent, post-quantum end of the spectrum [7]. The honest summary is a design space governed by a trade-off among three goods, succinctness, transparency, and prover efficiency, in which no single system maximises all three; Table 2 places representative systems within it.

Table 2. Representative Modern Proof Systems

System / family	Setup	Proof size	Post-quantum
Groth16 (pairing-based)	Trusted, per-circuit	Smallest	No
PLONK / Marlin (universal)	Trusted, universal + updatable	Small	No
Bulletproofs (IPA)	Transparent (none)	Small-to-medium	No
Halo-style (recursive)	Transparent (none)	Small	No
STARK (FRI, hash-based)	Transparent (none)	Large	Yes

IX. RECURSION, AGGREGATION, AND THE zkEVM

The application that is currently absorbing the most engineering effort, validity rollups, has pushed two further capabilities to the foreground. The first is recursion: a proof can attest to the validity of another proof, so that a single succinct proof can vouch for an entire chain of computations, and incrementally verifiable computation lets a long-running process maintain one constant-size proof of its whole history [18]. Recursion enables proof aggregation, combining many independent proofs into one that a verifier checks at the cost of a single proof, which is exactly what lets a rollup compress thousands of transactions into one on-chain verification [14], [18]. The second capability is generality. Early systems proved fixed, hand-built circuits; the ambition now is the zkEVM, a proof system for the full instruction set of a general smart-contract virtual machine, so that arbitrary existing contracts can be executed off-chain and proven correct without being rewritten as bespoke circuits [13], [14]. Realising this is formidable, because expressing a complete virtual machine as efficient constraints strains every part of the pipeline, and it is a major reason proving remains computationally expensive and why hardware acceleration of the prover has become an active field [7], [15]. Together, recursion and general-purpose proving are turning zero-knowledge proofs from a tool for specific statements into a general substrate for verifiable computation at scale.

X. CHALLENGES AND CONCLUSION

Significant challenges remain. Proving is computationally expensive: generating a proof can take orders of magnitude longer than performing the computation it attests to, which shifts cost from verifier to prover and motivates ongoing work on faster proving and hardware acceleration [7], [15]. Expressing a computation as the arithmetic circuit these systems require is intricate and error-prone, and a bug in the circuit can silently void the security guarantee, which has made better compilers and formal auditing a priority [16], [17]. For zk-SNARKs the trusted setup remains a structural risk that universal and updatable constructions reduce but do not fully remove, while for zk-STARKs the large proof size is the standing cost [11], [12].

Even so, the trajectory is unmistakable. A primitive that was a theoretical curiosity for three decades is now foundational infrastructure for privacy and scalability in distributed ledgers [1], [8], [14]. The two families embody a clean trade-off, zk-SNARKs minimising on-chain footprint at the price of a trusted setup and quantum exposure, zk-STARKs offering transparency and post-quantum security at the price of proof size, and active research is narrowing the gap from both directions [6], [7], [12]. As proving costs fall and tooling matures, the ability to prove a computation correct without revealing its inputs is becoming a general building block of distributed systems, well beyond the blockchains where it first found a compelling use.

REFERENCES

- [1] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on Computing*, vol. 18, no. 1, pp. 186–208, Feb. 1989.
- [2] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems," *Journal of the ACM*, vol. 38, no. 3, pp. 690–728, Jul. 1991.
- [3] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer, "From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again," in *Proc. 3rd Innovations in Theoretical Computer Science Conf. (ITCS)*, Cambridge, MA, USA, 2012, pp. 326–349.
- [4] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Succinct non-interactive zero knowledge for a von Neumann architecture," in *Proc. 23rd USENIX Security Symposium (USENIX Security)*, San Diego, CA, USA, 2014, pp. 781–796.

- [5] R. Gennaro, C. Gentry, B. Parno, and M. Raykova, "Quadratic span programs and succinct NIZKs without PCPs," in *Proc. 32nd Annual Int. Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, Athens, Greece, 2013, pp. 626–645.
- [6] J. Groth, "On the size of pairing-based non-interactive arguments," in *Proc. 35th Annual Int. Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, Vienna, Austria, 2016, pp. 305–326.
- [7] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Scalable, transparent, and post-quantum secure computational integrity," *IACR Cryptology ePrint Archive*, Rep. 2018/046, 2018.
- [8] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from Bitcoin," in *Proc. IEEE Symp. Security and Privacy (S&P)*, San Jose, CA, USA, 2014, pp. 459–474.
- [9] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," in *Proc. IEEE Symp. Security and Privacy (S&P)*, Berkeley, CA, USA, 2013, pp. 238–252.
- [10] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *Proc. IEEE Symp. Security and Privacy (S&P)*, San Francisco, CA, USA, 2018, pp. 315–334.
- [11] S. Bowe, A. Gabizon, and I. Miers, "Scalable multi-party computation for zk-SNARK parameters in the random beacon model," *IACR Cryptology ePrint Archive*, Rep. 2017/1050, 2017.
- [12] A. Gabizon, Z. J. Williamson, and O. Ciobotaru, "PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge," *IACR Cryptology ePrint Archive*, Rep. 2019/953, 2019.
- [13] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," Ethereum White Paper, 2014.
- [14] L. T. Thibault, T. Sarry, and A. S. Hafid, "Blockchain scaling using rollups: A comprehensive survey," *IEEE Access*, vol. 10, pp. 93039–93054, 2022.
- [15] J. Thaler, "Proofs, arguments, and zero-knowledge," *Foundations and Trends® in Privacy and Security*, vol. 4, nos. 2–4, pp. 117–660, 2022.
- [16] J. Eberhardt and S. Tai, "ZoKrates—Scalable privacy-preserving off-chain computations," in *Proc. IEEE Int. Conf. Blockchain*, Halifax, NS, Canada, 2018, pp. 1084–1091.
- [17] C. Reitwiessner, "zkSNARKs in a nutshell," Ethereum Foundation Technical Report, 2016.
- [18] S. Bowe, J. Grigg, and D. Hopwood, "Recursive proof composition without a trusted setup," *IACR Cryptology ePrint Archive*, Rep. 2019/1021, 2019.
- [19] B. Bünz, B. Fisch, and A. Szepieniec, "Transparent SNARKs from DARK compilers," in *Proc. 39th Annual Int. Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, Zagreb, Croatia, 2020, pp. 677–706.
- [20] A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in *Proc. 16th Int. Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT)*, Singapore, 2010, pp. 177–194.
- [21] A. Chiesa, Y. Hu, M. Maller, P. Mishra, N. Vesely, and N. Ward, "Marlin: Preprocessing zkSNARKs with universal and updatable SRS," in *Proc. 39th Annual Int. Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, Zagreb, Croatia, 2020, pp. 738–768.
- [22] J. Bootle, A. Cerulli, P. Chaidos, J. Groth, and C. Petit, "Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting," in *Proc. 35th Annual Int. Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, Vienna, Austria, 2016, pp. 327–357.