De[1]: Execution-Grounded Financial World Models from Replay, Imagination, and Live On-Chain Markets

De[1] Team

Abstract

World models are commonly trained on offline trajectories or purpose-built simulators, yet a deployed financial policy changes the state-action distribution on which it is evaluated. Static replay cannot reveal the market response to actions that were never taken, while simulated fills, impact, latency, and strategic response remain modeling assumptions. We present De[1], an execution-grounded architecture for federated financial world modeling. The system learns from historical on-chain behavior, generates action-conditioned latent rollouts for planning, and uses bounded live execution with finalized settlement evidence for calibration and adaptation. Raw intents, policies, and trajectories remain in participant-controlled domains; clipped and protected cohort updates improve shared dynamics. We formalize the executed-branch boundary, a finality-aware evidence object, risk-sensitive shadow planning, and a Stake-to-Train protocol that couples assigned work to pre-committed evaluation and objectively verifiable penalties. Replay supplies breadth, imagination supplies counterfactual hypotheses, and live markets supply the reality constraint.

1 Introduction

A useful financial world model is evaluated by its action-conditioned predictions, not by sequence prediction alone. An order enters an endogenous environment in which liquidity, fees, latency, inclusion, ordering, market impact, and other participants jointly determine the realized result. The evaluated policy can alter the environment that evaluates it.

Historical replay is scalable and indispensable, but the replayed market cannot respond to an action that was never taken. Learned and agent-based simulators permit intervention, but their fills, impact functions, latency, and strategic response are encoded assumptions. Live execution is therefore not a replacement for offline learning or model imagination. It is the grounding channel that keeps both answerable to the deployed environment.

Frontier work is converging on this closed-loop view. World-In-World finds that open-loop visual quality does not guarantee task success and that scaling action-observation post-training is more effective than merely upgrading the pretrained generator [8]. Across 31 environments, online world-model agents outperform offline counterparts; limited online interaction restores performance by supplying a self-correction mechanism under distribution shift [9]. DayDreamer learns directly on physical robots without a simulator, while offline world-model pretraining followed by real-world online fine-tuning enables few-shot adaptation to seen and unseen tasks [10, 11]. These results support a hybrid path: offline scale, model imagination, and bounded real-environment grounding.

LeCun's 2022 paper describes an architectural research program rather than a particular model with a measured training cost [4]. In physical domains, however, validating action-conditioned predictions often requires constructing or instrumenting a perception-action loop. V-JEPA 2 reduces environment-specific collection through internet-scale video pretraining and fewer than 62 hours of

unlabeled robot trajectories, yet still relies on a robot data and deployment pipeline for closed-loop validation [5]. We summarize the marginal grounding burden as

$$C_{\text{ground}} = C_{\text{environment}} + C_{\text{instrumentation}} + C_{\text{intervention}} + C_{\text{safety}}. \tag{1}$$

On-chain finance changes the first term: the market is not constructed for the model. Trading, routing, rebalancing, liquidation, and risk management already occur for independent economic reasons. With authorization and provenance, the learning system can reuse this production process. This does not make data free; instrumentation, execution cost, privacy, selection bias, and safe exploration remain material.

Let d_t^π denote the state-action-outcome distribution induced by the deployed policy at time t. The predictive risk that matters in deployment is

$$R_t^\pi(\Theta) = \mathbb{E}_{(h,a,y)\sim d_t^\pi}\left[\ell(f_\Theta(h,a),y)\right]. \tag{2}$$

Offline pretraining supplies broad priors; verified execution tuples provide direct evidence for calibrating Eq. (2) on the distribution the system actually encounters. This is predictive grounding, not automatic causal identification.

On-chain markets provide a useful first domain. Bitcoin established an economically ordered, independently checkable history of authorized transfers [1]; Ethereum generalized the ledger into a programmable state-transition system [2, 3]. Financial research shows why the response channel matters: market replay does not substantially adapt to an experimental strategy, interactive simulators inherit realism uncertainty, and real order-flow imbalance is associated with price change and market depth [12, 13]. Order-level foundation models such as MarS demonstrate that financial markets can be modeled as interactive virtual worlds [14]. A transaction receipt can attest that a branch executed under a declared finality policy. It cannot attest private intent, causal attribution, data rights, or model quality.

We make four system contributions:

1. an evidence model that binds world-model predictions to authorized, chain-finalized execution outcomes;
2. an action-conditioned latent world model with risk-sensitive shadow planning and explicit abstention outside empirical support;
3. a federated Stake-to-Train protocol that improves shared market dynamics without centralizing raw proprietary trajectories; and
4. an evaluation protocol that keeps realized outcomes, model-generated counterfactuals, and causal claims distinct.

Our thesis is architectural: the durable technical asset is not merely a larger dataset, but a continuing data-generating process. Replay provides memory, simulation provides imagination, and authorized live execution returns a new test of the model against reality. Together they form a data-sovereign training and calibration substrate for a financial world model.

2 Problem Formulation and Execution Evidence

2.1 State transitions and settlement evidence

Let σ_t denote a committed on-chain state and T_t an authorized transaction. Ethereum formalizes execution as

$$\sigma_{t+1} = \Upsilon(\sigma_t, T_t) \quad \text{or} \quad \text{ERROR}. \tag{3}$$

Transaction ordering can change realized economic value, especially under maximal extractable value and adversarial routing conditions. To bind an executed branch to a private decision context without publishing that context, the minimum evidence object is

$$\begin{aligned}\mathcal{E}_i = (\,&\mathrm{Com}(c_i;\omega_i), \mathrm{Com}(r_i;\nu_i),\\ &\mathrm{txHash}_i, \mathrm{receipt}_i, \mathrm{header}_i, \mathrm{stateRoot}_i,\\ &\pi_i^{\mathrm{auth}}, \pi_i^{\mathrm{final}}),\end{aligned} \tag{4}$$

where c_i is private intent and constraints, r_i is the proposed route, and the commitments are hiding and binding under fresh randomness. The authorization proof binds permissions, chain identity, model version, and the permitted action envelope; the finality proof binds the receipt to a canonical header under a declared chain policy. A chain-specific predicate $F_c(b, t)$ determines whether an episode is provisional, finalized, reverted, or ineligible for evaluation.

2.2 A partially observed endogenous market

We model the environment as a non-stationary, partially observed process:

$$p_t(s_{t+1}, y_t \mid h_t, a_t, \mu_t), \tag{5}$$

where h_t is the information history, a_t is an authorized action, y_t contains execution outcomes, and μ_t represents the evolving population of market participants and institutional rules. The time index is essential: deployment changes both the environment and the data distribution.

For episode i at domain k, we record

$$\tau_i^{(k)} = \left(c_i, \{o_t, a_t, e_t, y_t\}_{t=0}^{T_i-1}, o_{T_i}; m_i, \rho_i\right), \tag{6}$$

with mode $m_i \in \{\text{live}, \text{replay}, \text{synthetic}\}$ and signed provenance ρ_i. Modes are never merged silently.

2.3 Executed-branch boundary

Live execution observes the realized potential outcome for the selected action:

$$Y_t(a_t) \text{ is observed}, \qquad Y_t(a'),\ a' \neq a_t, \text{ is unobserved.} \tag{7}$$

Thus, settlement is a realized outcome under the deployed selection policy, not a complete counterfactual label and not an unconfounded causal estimate. Shadow rollouts may estimate $Y_t(a')$, but they do not convert it into market-verified fact.

3 System Architecture

The architecture contains two closed loops. The *reality loop* observes state, authorizes an action, routes and executes it, and confirms the result. The *learning loop* admits evidence, trains locally, aggregates protected updates, validates a candidate model, and publishes a version. The complete path is

$$\text{Replay} \rightarrow \text{Imagine} \rightarrow \text{Execute} \rightarrow \text{Calibrate} \rightarrow \text{Update}.$$

The data path, execution path, and learning path remain independently auditable state machines.

This separation prevents three common category errors. First, a quote or routing probe is not user intent. Second, newly captured telemetry is not evidence that weights have been updated or improved. Third, a cryptographic proof of computation does not establish scientific validity.

In this architecture, de1.exchange supplies an execution and telemetry interface to live on-chain markets. The existence of that data path does not establish the number of admissible episodes or the fidelity of the world model; both require the evaluation protocol in Section 6.

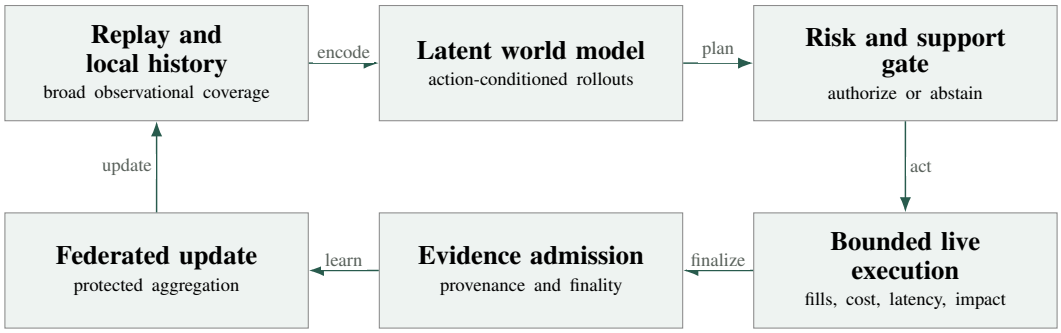

Figure 1: Execution-grounded world-model architecture. Replay provides coverage, latent rollouts support planning, and a risk gate limits real actions. Only finalized outcomes enter the federated learning path.

4 Execution-Grounded World Model

4.1 Latent dynamics and explicit outcomes

Following LeCun's objective-driven formulation [4] and action-conditioned latent prediction exemplified by V-JEPA 2 [5], the model predicts task-relevant representations rather than reconstructing every observable detail:

$$s_t = E_\phi(h_t), \tag{8}$$

$$s^+_{t+1} = \mathrm{sg}\big(E_{\bar{\phi}}(h_{t+1})\big)\,, \tag{9}$$

$$(\hat{s}_{t+1}, \hat{p}_\psi(y_t \mid s_t, a_t)) = M_\Theta(s_t, a_t, \xi_t). \tag{10}$$

Here ξ_t captures market responses not uniquely determined by current information. The latent state supports abstraction and long-horizon planning; the explicit distribution over y_t supports calibration, risk constraints, and audit.

The training objective is

$$\begin{aligned}\mathcal{L}_{\mathrm{WM}} = {} & d(\hat{s}_{t+1}, s^+_{t+1}) + \lambda_R \mathcal{R} \\ & - \beta \log p_\Theta(o_{t+1}, y_t \mid h_t, a_t) + \lambda_{\mathrm{cal}} \mathcal{L}_{\mathrm{cal}}.\end{aligned} \tag{11}$$

The observation-outcome term may decompose into proper scores for fill, slippage, latency, gas, impact, failure, and constraint events.

Following the agent-environment formulation of Silver et al. [20], De[1] treats intelligence as a consequence-bearing loop: an agent acts, the market responds, and the realized trajectory supplies task feedback. Here, "reward" is a risk-adjusted learning signal derived from execution outcomes:

$$\begin{aligned}r^{\mathrm{market}}_t = {} & U(\mathrm{fill}_t, -\mathrm{cost}_t, -\mathrm{latency}_t, -\mathrm{risk}_t) \\ & - \lambda_{\mathrm{con}} \mathbf{1}[\text{constraint breach}].\end{aligned} \tag{12}$$

The learning signal and the scientific evaluation target are therefore defined by realized market consequences, not by a proxy for participation or resource ownership.

Fei-Fei Li and collaborators emphasize spatial belief formation under partial observation and the transition from passive perception to active exploration [6, 7]. The financial analogue is not visual fidelity. It is preservation of decision-relevant dynamics, policy ranking, failure regions, and uncertainty.

4.2 Risk-sensitive shadow planning

Candidate sequences are rolled out inside the learned world:

$$\tilde{\tau}^{(m)} \sim p_\Theta(o_{t+1:t+H}, y_{t:t+H-1} \mid h_t, \boldsymbol{a}). \tag{13}$$

The planner selects

$$\begin{aligned}\boldsymbol{a}^* = \arg \min_{\boldsymbol{a} \in \mathcal{A}_{\mathrm{safe}} \cap \mathcal{A}_{\mathrm{support}}} & \big[\mathbb{E}_\Theta G(\boldsymbol{a}) + \lambda\, \mathrm{CVaR}_\alpha(G(\boldsymbol{a})) \\ & + \kappa U_\Theta(h_t, \boldsymbol{a})\big],\end{aligned} \tag{14}$$

where U_Θ is epistemic uncertainty and

$$G(\boldsymbol{a}) = \sum_{j=0}^{H-1} \gamma^j C(y_{t+j}). \tag{15}$$

Only the first authorized action is executed before replanning. The system abstains outside support or above an uncertainty threshold. Shadow outcomes and live outcomes remain separately labeled; a model-generated trajectory is never admitted as finalized market evidence.

5 Federated Training, Stake-to-Train, and Robust Validation

5.1 Local custody and protected updates

For K participant-controlled domains, raw intents, proprietary policies, and local trajectories remain in $\mathcal{D}_k$. This follows the original Federated Averaging separation between decentralized data and aggregated local updates [15]:

$$\min_{\Theta} F(\Theta) = \sum_{k=1}^{K} \alpha_k \mathbb{E}_{\tau \sim \mathcal{D}_k}[\mathcal{L}_{\mathrm{WM}}(\Theta; \tau)]. \tag{16}$$

Communication remains a first-order systems constraint. FedAvg reduced communication rounds by 10–100$\times$ relative to synchronized SGD in its evaluated settings [15]. Deep Gradient Compression reported 270–600$\times$ gradient compression across evaluated image, speech, and language workloads [18]; Auto-Precision Scaling demonstrated a complementary adaptive low-precision route [19]. Together, these results establish a practical engineering foundation for communication-efficient federated learning across heterogeneous financial clients.

In round r, each domain computes a clipped update

$$\Delta_k^c = \Delta_k \min\left(1, \frac{C}{\|\Delta_k\|_2}\right). \tag{17}$$

Error-feedback sparsification retains a local residual:

$$u_k^r = Q_r(\Delta_k^c + e_k^r), \qquad e_k^{r+1} = \Delta_k^c + e_k^r - u_k^r. \tag{18}$$

Compression optimizes bandwidth, while practical secure aggregation protects individual updates and tolerates stated dropout thresholds [16]. Cohort aggregation then adds robust cross-cohort validation:

$$g_m = \mathrm{SecAgg}\{u_k^r : k \in \mathcal{C}_m\}, \tag{19}$$

$$\bar{\Delta} = \mathrm{RobustAgg}(\{g_m\}_{m=1}^M; \{q_m\}), \tag{20}$$

$$\Theta^{r+1} = \Theta^r + \eta_r \bar{\Delta}. \tag{21}$$

Quality q_m is derived from provenance, reproducible holdout performance, calibration, and peer validation. It is independent of economic stake.

Optional client-level differential privacy adds

$$Z_r \sim \mathcal{N}(0, \sigma^2 C^2 I) \tag{22}$$

with a complete (ε, δ) accountant. Secure aggregation, homomorphic encryption, trusted execution, differential privacy, and on-chain provenance form a layered architecture for confidentiality, verifiability, and attribution across data, updates, and execution evidence.

5.2 Stake-to-Train: task-scoped accountable participation

Federated learning keeps raw data within participant-controlled domains; Stake-to-Train adds task-level accountability. We define *Stake-to-Train* as a task-scoped bonded protocol for trainers, data contributors, aggregators, and evaluators. A bond establishes eligibility for an assigned role and bounded liability for objectively verifiable protocol breaches, while scientific influence is determined exclusively by validated contribution.

Before enrollment, the task sponsor freezes a descriptor

$$\begin{aligned}\Gamma_J = \mathrm{Com}\big(&\Theta_J^0, \mathsf{Roles}_J, \mathsf{Schema}_J, \mathcal{L}_J, \mathcal{M}_J,\\ &H(\mathcal{B}_J), \mathcal{P}_J, t_J, \Omega_J, \ell_J, d_J\big),\end{aligned} \tag{23}$$

where Θ_J^0 is the initial model, $\mathcal{B}_J$ is the committed evaluation protocol, $\mathcal{P}_J$ is the privacy policy, Ω_J is a closed set of slashable violations, ℓ_J is the maximum slash, and d_J is the dispute window. Roles, schemas, objectives, metrics, and deadlines become immutable after enrollment.

Participant i accepts a role by signing the task commitment and posting a task-specific bond $b_{i,J}$. The participant then submits

$$W_{i,J} = \left(\mathrm{Com}(\Delta_{i,J}; s_{i,J}), \rho_{i,J}, \pi_{i,J}\right), \tag{24}$$

containing a committed update, provenance record, and verification evidence. Scientific evaluation remains independent of the bond:

$$\begin{aligned}q_{i,J} &= V(W_{i,J}; \mathcal{B}_J),\\ w_{i,J} &= A(q_{i,J}, n_{i,J}, \rho_{i,J}),\\ b_{i,J} &\notin \mathrm{Inputs}(V) \cup \mathrm{Inputs}(A).\end{aligned} \tag{25}$$

Here $n_{i,J}$ is capped admissible sample mass and A is the aggregation policy frozen in Γ_J. The bond determines task eligibility and accountability; validated work determines quality score, aggregation weight, benchmark result, and scientific conclusion.

Slashing activates only upon reproducibly provable violations:

$$s_{i,J} = \min\{b_{i,J}, \ell_J\}\mathbf{1}[\mathrm{Breach}(W_{i,J}, \Gamma_J, e_{i,J}) = 1], \qquad b_{i,J}^{\mathrm{return}} = b_{i,J} - s_{i,J}. \tag{26}$$

Examples include non-delivery, commitment mismatch, duplicate or stolen delivery, an invalid proof, or a signed protocol violation. Quality variation, non-IID updates, market outcomes, and scientific disagreement remain evaluation signals rather than penalty triggers. Every proposed slash carries reproducible evidence and remains challengeable during d_J.

Stake-to-Train builds on the FLock line of work on peer evaluation and malicious-participant defense [21, 22]. It complements statistical validation, robust aggregation, provenance checks, and privacy protection. Privacy-preserving per-client verification aligns individual accountability with secure aggregation.

5.3 Robust and verifiable aggregation

zkFL adds aggregation-integrity guarantees to this stack: it proves inclusion and commitment consistency for client updates together with faithful execution of the declared aggregation relation [23]. A verifiable De[1] round binds client commitments, the aggregate commitment, and the pre-declared rule A:

$$\begin{aligned}&\mathrm{Verify}(vk, C_{1:K}, C_{\bar{\Delta}}, \pi_r) = 1\\ &\Rightarrow \quad \bar{\Delta} = A(u_{1:K}) \ \wedge \ \bigwedge_{k=1}^{K} C_k = \mathrm{Com}(u_k; s_k).\end{aligned} \tag{27}$$

The proof establishes faithful execution of A over committed inputs. De[1] composes execution integrity with independent quality validation and privacy-preserving aggregation:

$$\begin{aligned}\mathcal{G}_{\mathrm{round}} &= \mathcal{G}_{\mathrm{integrity}} \wedge \mathcal{G}_{\mathrm{quality}} \wedge \mathcal{G}_{\mathrm{privacy}},\\ \mathcal{G}_{\mathrm{integrity}} &\leftarrow \mathrm{Verify}(\pi_r), \qquad \mathcal{G}_{\mathrm{quality}} \leftarrow V(W_{i,J}; \mathcal{B}_J),\\ \mathcal{G}_{\mathrm{privacy}} &\leftarrow \mathrm{SecAgg}(\{u_k^r\}) + \mathrm{DP}.\end{aligned} \tag{28}$$

This layered guarantee makes integrity, quality, and confidentiality jointly auditable within a single training round.

Table 1: Evaluation map from scientific claims to frozen evidence and decision-relevant metrics.

Claim	Required evidence	Metrics
Predictive fidelity	Frozen forecast and finalized outcome	NLL, Brier score, calibration, coverage
Decision fidelity	Matched shadow and live policies	Rank correlation, regret, CVaR, abstention
Adaptation	Forward-in-time regime splits	Degradation and recovery time
Federated value	Non-IID multi-domain splits	Mean, worst-domain, leave-one-domain-out
Security and privacy	Stress-tested client and aggregator behavior	Attack resistance, inversion exposure, membership exposure
Systems viability	End-to-end training rounds	Bytes, latency, dropout tolerance, proof overhead

6 Evaluation Protocol

6.1 Prequential and decision evaluation

Predictions are frozen before outcomes arrive. Finality-aware prequential negative log likelihood is

$$\mathrm{NLL}_{\mathrm{preq}} = -\frac{1}{|\mathcal{D}_{\mathrm{final}}|} \sum_{i \in \mathcal{D}_{\mathrm{final}}} \log p_{\theta_{i^-}}(o_{i,+}, y_i \mid h_i, a_i). \tag{29}$$

It must be stratified by chain, asset, order size, and regime, together with calibration and coverage. Decision relevance is evaluated using the agreement between shadow and live policy rankings:

$$\rho_{\mathrm{rank}} = \mathrm{Spearman}\left(\{\hat{J}_{\mathrm{shadow}}(\pi_j)\}, \{J_{\mathrm{live}}(\pi_j)\}\right). \tag{30}$$

When safe randomization is unavailable, policy comparison reports propensity overlap, effective sample size, weight clipping, confidence intervals, and a doubly robust estimate.

Core ablations compare replay only; replay plus latent imagination; replay, imagination, and execution grounding; centralized versus federated training; uncompressed versus compressed updates; and standard versus adversarial clients. All model-selection decisions are made on data preceding the frozen evaluation window.

6.2 Deployment and scientific evidence

A production execution path demonstrates an operational data route under real liquidity, latency, ordering, and economic cost. Chain coverage, transaction scale, and long-running uptime characterize deployment breadth and continuity. The prequential protocol complements these operational measures with predictive calibration, decision quality, robustness, and forward-in-time adaptation.

7 Conclusion

Financial intelligence becomes action-aware when it predicts how liquidity, price, execution, and risk evolve in response to decisions and then learns from realized consequences. De[1] combines market history as memory, latent rollouts as imagination, and bounded live execution as calibration. Finalized outcomes ground the executed branch; Stake-to-Train turns federated participation into task-scoped, verifiable responsibility; protected aggregation converts accepted local work into shared model improvement while proprietary context remains within participant-controlled domains.

On-chain finance already operates as a global, continuously evolving environment for human and machine financial activity. Replay supplies breadth, imagination explores possible futures, and live execution supplies reality. The model meets the market, and the market closes the learning loop.

References

[1] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008. Paper.

[2] V. Buterin, *A Next-Generation Smart Contract and Decentralized Application Platform*, Ethereum Whitepaper, 2014. Paper.

[3] G. Wood, *Ethereum: A Secure Decentralised Generalised Transaction Ledger*, Ethereum Yellow Paper, 2014. Paper.

[4] Y. LeCun, *A Path Towards Autonomous Machine Intelligence*, OpenReview, 2022. Paper.

[5] M. Assran et al., *V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning*, 2025. Project page.

[6] J. Yang et al., *Thinking in Space: How Multimodal Large Language Models See, Remember, and Recall Spaces*, arXiv:2412.14171, 2024.

[7] P. Zhang et al., *Theory of Space: Can Foundation Models Construct Spatial Beliefs through Active Exploration?*, ICLR, 2026.

[8] J. Zhang et al., *World-In-World: World Models in a Closed-Loop World*, ICLR Oral, 2026. Paper.

[9] J. Chen et al., *Offline vs. Online Learning in Model-based RL: Lessons for Data Collection Strategies*, RLC, 2025. Paper.

[10] P. Wu et al., *DayDreamer: World Models for Physical Robot Learning*, CoRL, 2023. Paper.

[11] Y. Feng et al., *Finetuning Offline World Models in the Real World*, CoRL, 2023. Paper.

[12] T. H. Balch et al., *How to Evaluate Trading Strategies: Market Replay or Interactive Simulation?*, ICML Workshop on AI in Finance, 2019. Paper.

[13] R. Cont, A. Kukanov, and S. Stoikov, *The Price Impact of Order Book Events*, Journal of Financial Econometrics, 2014. Paper.

[14] J. Li et al., *MarS: A Financial Market Simulation Engine Powered by Generative Foundation Model*, ICLR, 2025. Paper.

[15] H. B. McMahan et al., *Communication-Efficient Learning of Deep Networks from Decentralized Data*, AISTATS, PMLR 54:1273–1282, 2017. Paper.

[16] K. Bonawitz et al., *Practical Secure Aggregation for Privacy-Preserving Machine Learning*, ACM CCS, pp. 1175–1191, 2017. DOI.

[17] L. Zhu, Z. Liu, and S. Han, *Deep Leakage from Gradients*, NeurIPS, vol. 32, 2019. Paper.

[18] Y. Lin et al., *Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training*, ICLR, 2018. Paper.

[19] R. Han, J. Demmel, and Y. You, *Auto-Precision Scaling for Distributed Deep Learning*, ISC High Performance, pp. 79–97, 2021. Paper.

[20] D. Silver et al., *Reward is enough*, Artificial Intelligence, vol. 299, art. 103535, 2021. DOI.

[21] N. Dong et al., *FLock: Defending Malicious Behaviors in Federated Learning with Blockchain*, NeurIPS DMLR Workshop, 2022.

[22] N. Dong et al., *Defending Against Poisoning Attacks in Federated Learning with Blockchain*, IEEE Trans. Artif. Intell., 2024.

[23] Z. Wang et al., *zkFL: Zero-Knowledge Proof-Based Gradient Aggregation for Federated Learning*, IEEE Trans. Big Data, 11(2):447–460, 2025. DOI.