DiSHE: Dispersion-Guided Selective Homomorphic Encryption for Federated Learning

Anonymous authors
Paper under double-blind review

Abstract

Homomorphic encryption (HE) lets a federated learning server aggregate client updates that it cannot read, but encrypting every parameter is expensive. Selective HE therefore encrypts only part of each update, usually under a single mask that all clients agree on. We present DiSHE, a selective CKKS protocol in which each client chooses its own mask. Updates are cut on a public grid of ciphertext-aligned chunks, so masks that differ across clients still aggregate slot by slot. Clients rank chunks by *chunk gradient dispersion* (CGD), the per-coordinate sample variance of per-example gradients on a small round-start probe, and always encrypt the classifier head. The same scores define a family of aggregation rules between FedAvg and inverse-dispersion weighting. Because encryption leaves the aggregated values unchanged, the learned model depends, in exact arithmetic, only on the aggregation rule; we characterize the resulting family, including a lower bound on every client's weight, and bound the ciphertext part of the stateless broadcast. With ResNet-18, five non-IID clients, ten rounds, and a nominal encryption ratio of 10%, the final global test accuracy of DiSHE is within 0.83 points of MaskCrypt and FedML-HE on CIFAR-10 and SVHN, and within 2.30 points on Fashion-MNIST. It exceeds our clipping-and-noise versions of FAS and ParaAegis by 4.96–49.51 points. At a 10% ratio, DiSHE needs no mask exchange and has the lowest per-round protocol time (1.51 s versus 1.82–15.54 s) of the five implementations. In masked-gradient attacks on LeNet-5, DiSHE is the only method that blocks both analytic label attacks, iDLG and iLRG, in every attempt.

1 Introduction

Federated learning (FL) trains a shared model on data that never leaves its owners: clients optimize locally and send only updates to a server that aggregates them (McMahan et al., 2017; Kairouz et al., 2021). The updates are nevertheless a side channel. Optimization-based attacks reconstruct training images from gradients (Zhu et al., 2019; Geiping et al., 2020; Yin et al., 2021), even from multi-step local updates (Dimitrov et al., 2022), and labels follow analytically from the classifier gradient (Zhao et al., 2020; Wainakh et al., 2022; Ma et al., 2023). Homomorphic encryption (HE) lets the server aggregate encrypted updates without seeing or perturbing them (Phong et al., 2018; Zhang et al., 2020), and CKKS (Cheon et al., 2017) makes this efficient for packed real-valued vectors. The price is size. Hu & Li (2025) measure CKKS ciphertexts at about twenty times the size of the values they pack, so *selective* HE encrypts only part of each update (Section 2).

Hu & Li (2025, Sec. 3.1) reject per-client masks for two reasons. Under CKKS batching, masks that differ across clients place different weights in the same ciphertext slots, and every coordinate that any client encrypts becomes a ciphertext in the aggregate. The first problem comes from the way coordinates are packed and disappears with a ciphertext-aligned layout. The second is a real cost, which we bound. A further observation shapes our design. Encryption does not change the aggregated values, so in a protocol without noise the learned model depends only on the aggregation rule, while the choice of encrypted coordinates affects only the cost and what the server sees. Which coordinates to hide and how to weight clients can therefore be studied separately.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

DiSHE builds on both observations. Clients cut their updates on a fixed public grid of chunks, one chunk per ciphertext, and each client ranks its chunks by *chunk gradient dispersion* (CGD), the per-coordinate sample variance of per-example gradients on a small probe drawn at the start of the round. The server reuses the scores to weight the aggregate. Our contributions are:

- **Selective CKKS without mask consensus.** A ciphertext-aligned chunk grid, a scale-invariant allocation of the budget across layers, a mandatory classifier head, and a stateless server whose ciphertext broadcast is bounded by the union of the current masks (Proposition 2). The protocol still assumes a public chunk map, a shared key, and synchronized clients.
- **A dispersion score for selection**, computed from two per-chunk sufficient statistics and with a leave-one-out interpretation (Proposition 4).
- **A family of aggregation rules and its analysis.** The weights $\lambda(\gamma)$ run from FedAvg ($\gamma = 0$) to inverse-dispersion weighting ($\gamma = 1$, the rule we implement and evaluate). Because encryption does not change the learned model (Remark 1), the family isolates the effect of the weighting: every $\gamma < 1$ keeps each client's weight at least $(1 - \gamma)$ times its FedAvg weight, and the aggregate interpolates between the two ends (Proposition 1) at the same encryption cost.
- **An evaluation against four selective-HE baselines**, of accuracy on three datasets and of cost at six encryption ratios, read through this family: noise-free baselines with data-size weights approximate its $\gamma = 0$ end, and DiSHE is its $\gamma = 1$ end.

DiSHE combines consensus-free, stateless selective HE with an encrypted classifier head, strongly overlapping client masks, and the lowest measured protocol time of the five implementations. At a 10% client budget, the aggregate encrypts just 16.2% of chunks. Final global accuracy is within one percentage point of MaskCrypt and FedML-HE on CIFAR-10 and SVHN.

2 Related Work

Leakage and defenses. Gradient inversion matches dummy gradients to observed ones with Euclidean (Zhu et al., 2019) or cosine objectives and image priors (Geiping et al., 2020), recovers batches from batch-normalization statistics (Yin et al., 2021), and extends to multi-step updates (Dimitrov et al., 2022); labels leak from the classifier gradient (Zhao et al., 2020; Wainakh et al., 2022; Ma et al., 2023), and membership inference is a further threat (Shokri et al., 2017; Nasr et al., 2019). Differential privacy (DP) clips and perturbs updates at a cost in accuracy (Abadi et al., 2016; McMahan et al., 2018), secure aggregation reveals only their sum (Bonawitz et al., 2017), hybrids combine the two (Truex et al., 2019), and multiparty HE removes the shared key (Mouchet et al., 2021).

Selective HE. FedML-HE fixes one mask before training from aggregated sensitivity maps (Jin et al., 2023); MaskCrypt merges gradient-guided client proposals into a mask every round and returns decrypted aggregates to the server (Hu & Li, 2025); FAS encrypts a fixed random fraction and perturbs and scrambles the rest (Korkmaz & Rao, 2025); and ParaAegis encrypts a voted set of large-norm coordinates and adds noise to the clipped remainder (Wu et al., 2025). DiSHE needs no mask agreement, perturbs nothing, and ranks chunks by the spread of per-example gradients, which has been studied as gradient diversity (Yin et al., 2018) and through the gradient noise scale (McCandlish et al., 2018).

3 Problem Setup and Threat Model

Federated training. N clients hold datasets $\mathcal{D}_i$ of sizes n_i and train $\theta \in \mathbb{R}^d$ under the cross-entropy loss $\ell(\theta; x, y)$, all of them in every round. In round t each client loads the common model θ^t, resets its Adam optimizer (Kingma & Ba, 2015), and runs E epochs of mini-batch Adam to obtain the update $\Delta_i^t = \theta_i^{t,\mathrm{loc}} - \theta^t$. FedAvg weights clients by $a_i = n_i / \sum_r n_r$ (McMahan et al., 2017).

Encryption and threat model. Clients share a CKKS secret key; the server holds a context without it. A ciphertext has $S = 4096$ slots at ring degree 8192, and the protocol needs only additions of

ciphertexts and plaintexts ($\boxplus$) and ciphertext–scalar products ($\boxdot$) with one rescaling. The server is *honest-but-curious*: it follows the protocol, stores every message, and attacks offline, knowing the algorithm, the public chunk map, each n_i, the initial model, and every plaintext chunk, ciphertext, flag, and score it receives. It holds no secret key and does not collude with the honest clients; malicious servers (Fowl et al., 2022), collusion, and key compromise are out of scope. Our goal is to limit what the server can reconstruct about training inputs and labels when a prescribed fraction of chunks is encrypted, without any agreement on masks. DiSHE gives no DP guarantee, and CKKS does not hide plaintext chunks, flags, or scores.

4 DiSHE

Figure 2 in Appendix C illustrates one round, Algorithms 1 to 3 give the complete procedure, and Appendix D gives implementation details.

4.1 Ciphertext-aligned chunks

A *logical layer* groups the trainable tensors of one layer: a convolution together with the affine parameters of the group normalization that follows it, or a fully connected layer with its bias. Group normalization (Wu & He, 2018) keeps no statistics of client data that could serve as reconstruction priors (Yin et al., 2021) and is more robust to label skew than batch normalization (Hsieh et al., 2020). The tensors of layer l are concatenated in a fixed order into $v_l \in \mathbb{R}^{d_l}$, which is split into $\lceil d_l/b \rceil$ consecutive chunks of at most b values, where b is the *logical chunk length*. A chunk may cross tensors but never layers. The chunk set $\mathcal{C} = \bigcup_l \mathcal{C}_l$, with $C = |\mathcal{C}|$, and its map to tensor offsets are public. We write $\Delta_{i,c}$ for an update restricted to chunk c, which has $d_c \le b$ real coordinates, and $\mathrm{pad}_b(\cdot)$ for zero-padding to length b. Padding never enters the statistics or the model. Each encrypted chunk occupies its own ciphertext ($b \le S$): a full ResNet-18 chunk ($b = 4096$) fills it, whereas LeNet-5 chunks ($b = 256$) are not packed together. Because chunk c occupies the same slots for every client, the server can add ciphertexts and plaintexts of the same chunk slot by slot. This removes the misalignment that led Hu & Li (2025, Sec. 3.1) to reject per-client masks, at the cost of partly empty ciphertexts when $b < S$.

4.2 Round-start chunk gradient dispersion

At the start of round t, before local training, client i draws a probe $\mathcal{B}_i^t \subseteq \mathcal{D}_i$ of $m_i = \min(512, n_i)$ examples uniformly without replacement, with a seed derived from the experiment seed, the round, and the client. It evaluates per-example gradients $g_{i,c,j} = \nabla_{\theta_c} \ell(\theta^t; x_j, y_j)$ at the incoming model θ^t, which is held fixed, and accumulates $G_{i,c}^t = \sum_{j \in \mathcal{B}_i^t} g_{i,c,j}$ and $Q_{i,c}^t = \sum_{j \in \mathcal{B}_i^t} \|g_{i,c,j}\|_2^2$ in micro-blocks, so the per-example gradients of the whole probe are never stored at once.

Definition 1 (Chunk gradient dispersion)**.** For $m_i \ge 2$, with $\bar{g}_{i,c} = G_{i,c}^t/m_i$ and the equality holding in exact arithmetic,

$$\widehat{D}_{i,c}^t = \max\left\{0,\ \frac{Q_{i,c}^t - \left\|G_{i,c}^t\right\|_2^2/m_i}{(m_i-1)\,d_c}\right\} = \frac{1}{(m_i-1)\,d_c} \sum_{j \in \mathcal{B}_i^t} \|g_{i,c,j} - \bar{g}_{i,c}\|_2^2 ; \tag{1}$$

for $m_i = 1$ we set $\widehat{D}_{i,c}^t = 0$.

CGD is the trace of the sample covariance of the per-example gradients on chunk c, divided by d_c; for $m_i \ge 2$ and in exact arithmetic it is unbiased for the same quantity over the whole local dataset. It measures the dispersion of gradient vectors rather than of their norms, and variation rather than information: a chunk whose per-example gradients coincide has zero CGD, although its update depends on the data.

4.3 Budget, allocation, and selection

Budget and mandatory head. Let $\mathcal{M}$ be the chunks of the final classifier. For $\alpha \in (0, 1]$, every client encrypts exactly K chunks, where

$$K = \max\big(\mathrm{rnd}(\alpha C),\ |\mathcal{M}|\big), \qquad K_{\mathrm{free}} = K - |\mathcal{M}|, \qquad \alpha_{\mathrm{eff}} = K/C, \tag{2}$$

and rnd rounds to the nearest integer with ties to even; for $\alpha = 0$, $K = 0$. The head is always encrypted because its gradient reveals labels, both for single examples (Zhao et al., 2020) and for batches (Wainakh et al., 2022; Ma et al., 2023). It is also small: 2 of the 2,746 chunks of our ResNet-18 and 4 of the 243 chunks of our LeNet-5. A constant K hides how many chunks a client encrypts but not which ones, since the flags reveal them.

Allocation and selection. With free chunks $\mathcal{F}_l = \mathcal{C}_l \setminus \mathcal{M}$, client i weights layer l by the effective count of its scores (Kish, 1965),

$$\omega_{i,l} = N_{\text{eff}}\big((\widehat{D}^t_{i,c})_{c\in\mathcal{F}_l}\big), \qquad N_{\text{eff}}(\boldsymbol{x}) = \frac{\big(\sum_c x_c\big)^2}{\sum_c x_c^2} \quad (N_{\text{eff}}(\mathbf{0}) = 0), \tag{3}$$

and splits K_{free} among layers in proportion to these weights, using a capacity-aware largest-remainder rule that places exactly K_{free} units without exceeding the size of any layer (Algorithm 3). N_{eff} equals k when k chunks are equally dispersed, approaches one when a single chunk dominates, and is scale-invariant (Lemma 1). A layer's share therefore reflects how evenly its dispersion is spread, not its gradient scale, which varies across layers by orders of magnitude. Within each layer, the client takes the $q_{i,l}$ free chunks with the largest CGD, breaking ties by global index:

$$\mathcal{E}^t_i = \mathcal{M} \cup \bigcup_l \mathrm{Top}_{q_{i,l}}\big\{\widehat{D}^t_{i,c} : c \in \mathcal{F}_l\big\}, \qquad |\mathcal{E}^t_i| = K. \tag{4}$$

4.4 Payload and dispersion-aware aggregation

In the public order, client i sends $\mathsf{Enc}(\mathrm{pad}_b(\Delta^t_{i,c}))$ for each chunk $c \in \mathcal{E}^t_i$ and the padded plaintext vector for every other chunk, together with a Boolean encryption flag and the float32 score $\widehat{D}^t_{i,c}$ of every chunk, encrypted or not. The server then computes, for each chunk,

$$\varepsilon^t_c = \max\big(\mathrm{median}_i\, \widehat{D}^t_{i,c},\, 10^{-12}\big), \quad \pi^t_{i,c} = \frac{n_i}{\widehat{D}^t_{i,c} + \varepsilon^t_c}, \quad w^t_{i,c} = \frac{\pi^t_{i,c}}{\sum_r \pi^t_{r,c}}, \tag{5}$$

with the median taken across clients, and forms the weights $\lambda^t_{i,c}(\gamma) = (1-\gamma)\, a_i + \gamma\, w^t_{i,c}$ for $\gamma \in [0,1]$. If client updates estimated a common quantity with variances proportional to $\widehat{D}_{i,c}/n_i$, inverse-variance weighting would use $n_i/\widehat{D}_{i,c}$ (Cochran, 1954), and the median offset stabilizes weights built from estimated variances (Cochran & Carroll, 1953); since these conditions are not established for non-IID clients running Adam, we treat w as a stabilized heuristic. When the median score of a chunk is near zero, w can concentrate on low-dispersion clients and nearly silence the others (Remark 3 in Appendix B). The index γ thus defines a family DiSHE$_\gamma$ from FedAvg ($\gamma = 0$) to pure inverse-dispersion weighting ($\gamma = 1$). Throughout, DiSHE without a subscript denotes DiSHE$_1$, the configuration we implement and evaluate; intermediate values of γ are analyzed in Section 5 but not run. With $\mathcal{H}^t_c$ and $\mathcal{V}^t_c$ the clients that sent c encrypted and in plaintext, the server forms

$$U^t_c = \Big(\boxplus_{i\in\mathcal{H}^t_c} \lambda^t_{i,c} \boxdot \mathsf{ct}_{i,c}\Big) \boxplus \sum_{i\in\mathcal{V}^t_c} \lambda^t_{i,c}\, \mathrm{pad}_b(\Delta^t_{i,c}), \tag{6}$$

which is a ciphertext if $\mathcal{H}^t_c \neq \emptyset$ and a plaintext otherwise. Each encrypted term costs one ciphertext–scalar product on a fresh ciphertext, so the multiplicative depth is one, and no client value is dropped or perturbed.

4.5 Stateless broadcast and the server's visible-update estimate

The server keeps no model. It broadcasts U^t with its flags, and each client decrypts the ciphertext chunks, strips the padding, and sets $\theta^{t+1} = \theta^t + U^t$, so all clients hold the same model and no decrypted value reaches the server, unlike in MaskCrypt, where clients return decrypted aggregates to the server (Hu & Li, 2025). The server can still track the model through what it sees in the clear. Its *visible-update estimate* adds every plaintext contribution, including those to chunks whose aggregate is encrypted, and omits the encrypted ones:

$$\hat\theta^0 = \theta^0, \qquad \hat\theta^{t+1}_c = \hat\theta^t_c + \textstyle\sum_{i\in\mathcal{V}^t_c} \lambda^t_{i,c}\, \Delta^t_{i,c}. \tag{7}$$

This extends the exposed weights of MaskCrypt to the global model; it is not an optimal estimator, and flags, scores, and history could support stronger attacks.

5 ANALYSIS

Proofs are in Appendix B. Throughout this section, arithmetic is exact: we ignore CKKS approximation error and float32 storage (Appendix F), and the clamp in Equation (1) is then inactive.

CGD has a simple reading (Proposition 4 in Appendix B): for a hypothetical gradient step on the probe mean, it is proportional to the mean squared per-coordinate displacement caused by removing one probe example. This motivates the score but says nothing about the transmitted multi-step update, and it does not show that high-CGD chunks are the ones an attacker needs most (Section 6.4).

Remark 1 (Encryption does not change the learned model). Decrypting Equation (6) gives $\sum_i \lambda_{i,c}\Delta_{i,c}$ however the clients are split between $\mathcal{H}_c$ and $\mathcal{V}_c$. For fixed data, seeds, and γ, the global models of DiSHE$_\gamma$ are therefore the same for every ratio α and every selection rule, and a protocol that aggregates unperturbed updates with data-size weights follows DiSHE$_0$ when data, initialization, and local training coincide. Accuracy thus carries no information about the selection rule, which affects only cost and what the server sees. MaskCrypt and FedML-HE approximate the $\gamma = 0$ end in our runs, which do not certify identical data and initialization (Appendix G), and DiSHE is the $\gamma = 1$ end.

Proposition 1 (The aggregation family). *Fix a round, the incoming model θ, the updates Δ_i, and the scores, and let $x_\gamma = \theta + U(\gamma)$ with $U_c(\gamma) = \sum_i \lambda_{i,c}(\gamma)\Delta_{i,c}$. For every chunk c and $\gamma \in [0,1]$: (a) $\lambda_{\cdot,c}(\gamma)$ is a probability vector with $(1-\gamma)a_i \le \lambda_{i,c}(\gamma) \le (1-\gamma)a_i + \gamma$, and it equals a when $\gamma = 0$ or when all clients report the same score; (b) $x_\gamma = (1-\gamma)x_0 + \gamma x_1$ and $\|x_1 - x_0\|^2 \le \sum_c \kappa_c^2 r_c^2$, where $\kappa_c = \sum_i |w_{i,c} - a_i| \le 2$ and $r_c = \max_i \|\Delta_{i,c} - U_c(0)\|_2$; (c) if $F : \mathbb{R}^d \to \mathbb{R}$ has an L-Lipschitz gradient on the segment $[x_0, x_1]$, then*

$$\left|F(x_\gamma) - (1-\gamma)F(x_0) - \gamma F(x_1)\right| \le \frac{L}{2}\,\gamma(1-\gamma)\,\|x_1 - x_0\|^2 . \tag{8}$$

Part (a) is the only lower bound on a client's weight: at $\gamma = 1$ a client can receive a negligible weight on a chunk (Remark 3), whereas every $\gamma < 1$ keeps it at least $(1-\gamma)a_i$. Parts (b) and (c) say that within a round the family moves along a segment whose length is set by how much clients disagree, and that any smooth objective stays within $\frac{L}{8}\|x_1 - x_0\|^2$ of the interpolated endpoint values; across rounds, the trajectory is stable in γ under a Lipschitz assumption (Proposition 5 in Appendix B). None of this bounds accuracy (Remark 4), so comparable accuracy at intermediate γ remains a hypothesis, motivated by the endpoint results of Section 6.2.

Remark 2 (Cost does not depend on γ). Encrypted sets depend on the scores, not on the weights, so ciphertexts, bytes, and client computation are the same for every γ; at $\gamma = 0$ the score upload could also be dropped.

Proposition 2 (Per-round ciphertext downlink). *The ciphertext chunks of U^t are exactly $\mathcal{U}^t = \bigcup_i \mathcal{E}_i^t$. For $\alpha > 0$,*

$$K \;\le\; |\mathcal{U}^t| \;\le\; \min\big\{C,\; |\mathcal{M}| + N(K - |\mathcal{M}|)\big\}, \tag{9}$$

and $\mathcal{U}^t = \emptyset$ for $\alpha = 0$. A server that instead kept an encrypted global state would hold ciphertexts on $\bigcup_{s \le t} \mathcal{U}^s$, which is non-decreasing in t.

The bound counts ciphertext chunks, not bytes or protection, and holds for any dependence between the masks; for ResNet-18 at $\alpha = 0.10$ it gives $275 \le |\mathcal{U}^t| \le 1{,}367$ of 2,746. Independent uniform masks would give $|\mathcal{M}| + (C - |\mathcal{M}|)\big[1 - \big(1 - \frac{K - |\mathcal{M}|}{C - |\mathcal{M}|}\big)^N\big] \approx 1{,}121$ in expectation (Table 4); the measured union is much smaller (Section 6.3).

Proposition 3 (Visible-update estimate). *For every chunk c and round t, $\theta_c^t - \hat{\theta}_c^t = \sum_{s<t}\sum_{i \in \mathcal{H}_c^s} \lambda_{i,c}^s\,\Delta_{i,c}^s$. Hence $\hat{\theta}_c^t = \theta_c^t$ if no client encrypted c before round t, and the recursion in Equation (7) never adds an encrypted contribution.*

The gap need not grow, since hidden contributions can cancel, and an omitted contribution may still be inferable later. For $\alpha > 0$ the head estimate stays at its initial value; in round 0 the estimate is exact.

6 EXPERIMENTS

6.1 SETUP

Runs. All experiments use five fully participating clients, a Dirichlet label partition with concentration 0.5, $E = 5$ epochs of Adam (learning rate 10^{-3}, batch size 128), and a 512-example probe, and each covers DiSHE and the four baselines (Appendix G). The accuracy runs train ResNet-18-GN on CIFAR-10, Fashion-MNIST, and SVHN at $\alpha = 0.10$ for ten rounds with seeds 123, 456, and 789; the cost runs execute three rounds of ResNet-18-GN on CIFAR-10 with seed 123 at ratios from 0 to 100%; and the attack runs train LeNet-5 on Fashion-MNIST at $\alpha = 0.05$ for ten rounds with the three seeds.

Data, models, and encryption. CIFAR-10 (Krizhevsky, 2009) and SVHN (Netzer et al., 2011) ($32 \times 32 \times 3$) and Fashion-MNIST (Xiao et al., 2017) ($28 \times 28 \times 1$) are scaled to $[0, 1]$, without augmentation. ResNet-18-GN is the CIFAR variant of ResNet-18 (He et al., 2016) with group normalization; at $b = 4096$ it has $C = 2{,}746$ chunks in 21 logical layers, and LeNet-5 (LeCun et al., 1998) has $C = 243$ chunks at $b = 256$ (Appendix E). Each client receives a fixed number of examples, 10,000 in the CIFAR-10 accuracy runs and 2,000 in all other runs, from a Dirichlet allocation (Hsu et al., 2019) that is adjusted to these sizes and can assign an example to more than one client (Appendix G). At $\alpha = 0.10$ a client encrypts $K = 275$ chunks, two of them the head. We use CKKS as implemented in TenSEAL (Benaissa et al., 2021), with ring degree 8192, coefficient moduli of $[60, 30, 60]$ bits, and scale 2^{30}. The 150-bit total coefficient modulus is within the 218-bit maximum specified by the HE standard for 128-bit classical security at ring degree $N = 8192$, under its assumed secret and error distributions (Albrecht et al., 2021); this verifies the encryption-parameter security level rather than the privacy of the overall protocol.

Baselines. We reimplemented four selective-HE methods in our framework, sharing the model, the data-loading and local-training code, and the seeds (Appendix H). **MaskCrypt** (Hu & Li, 2025) merges gradient-guided proposals of the clients into a common mask every round; we run a shared-key variant instead of its per-client-key protocol. **FedML-HE** (Jin et al., 2023) fixes a mask before training from aggregated sensitivity maps, with a simplified sensitivity score. Both aggregate with data-size weights and perturb nothing. Our **FAS** adaptation (Korkmaz & Rao, 2025) encrypts a seeded random coordinate mask fixed across clients and rounds and bitwise-scrambles the rest of the update, and our **ParaAegis** implementation (Wu et al., 2025) votes every round on the coordinates with the largest update magnitude. Both clip the unencrypted part to norm 1 and add Gaussian noise (the published FAS uses Laplace noise) whose scale follows a ParaAegis-style formula with nominal $(\epsilon, \delta) = (1, 10^{-5})$, which is not a verified DP guarantee for our versions (Appendix H). All methods use a nominal ratio of 0.10, but DiSHE counts chunks and the baselines count coordinates, so ciphertexts, bytes, and protection are not matched.

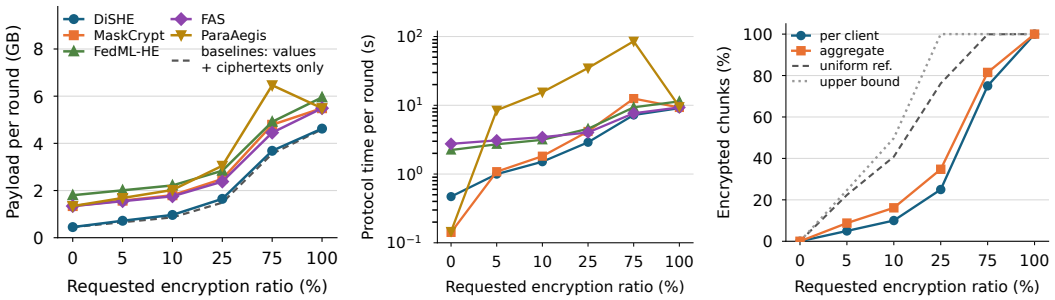
Metrics. The primary metric is the *final global test accuracy*, that is, the test accuracy of the decrypted model after round ten, reported as mean $\pm$ sample standard deviation over the three seeds. As a secondary description of local fitting, we report the *mean best-round local training accuracy* (Appendix I). With three seeds, all comparisons are descriptive.

6.2 UTILITY

Noise-free methods and the aggregation family. DiSHE achieves accuracy close to MaskCrypt and FedML-HE on CIFAR-10 and SVHN, with final global test accuracy differing by less than one percentage point (Table 1). On Fashion-MNIST, its mean accuracy is 2.30 and 1.77 points lower, respectively, with greater variation across seeds. In the terms of Section 5, MaskCrypt and FedML-HE approximate the $\gamma = 0$ end of the family and DiSHE is its $\gamma = 1$ end, although uncertified differences in data and initialization prevent attributing the gaps to the weighting alone. The two approximations of the $\gamma = 0$ end differ by up to 0.61 points, and the $\gamma = 1$ end lies between them on SVHN and 0.35 points below the lower one on CIFAR-10; three seeds cannot establish equivalence, and intermediate γ remains untested. On Fashion-MNIST, where the $\gamma = 1$ end is lower, a smaller γ is a natural candidate, since it keeps each client's weight at least $(1 - \gamma)a_i$ at the same cost (Proposition 1(a), Remark 2). Among the noise-free methods, the remaining differences lie in cost, coordination, and what the server observes (Sections 6.3 and 6.4).

Table 1: Accuracy (%) with ResNet-18-GN at a nominal 10% ratio. Left: final global test accuracy, mean ± sample deviation over three seeds. Right: mean best-round local training accuracy, ± the deviation across the five client-index means, not across runs (Appendix I). FAS and ParaAegis are our clipping-and-noise versions (Section 6.1); Figure 3 in Appendix I plots these values.

	Final global test accuracy			Mean best-round local training accuracy		
Method	CIFAR-10	F-MNIST	SVHN	CIFAR-10	F-MNIST	SVHN
DiSHE	73.00 ± 1.73	81.35 ± 4.15	84.17 ± 0.83	94.50 ± 0.95	96.78 ± 0.72	96.27 ± 0.66
MaskCrypt	73.35 ± 0.63	83.64 ± 2.46	84.68 ± 0.96	95.56 ± 1.04	97.01 ± 0.99	96.96 ± 0.79
FedML-HE	73.83 ± 0.79	83.12 ± 1.74	84.07 ± 0.94	94.70 ± 1.24	96.61 ± 0.93	95.58 ± 0.88
FAS	32.74 ± 1.56	60.63 ± 1.34	34.66 ± 8.56	70.71 ± 2.00	89.16 ± 2.94	80.01 ± 1.30
ParaAegis	64.74 ± 2.99	76.38 ± 1.79	70.77 ± 5.84	86.21 ± 1.32	93.58 ± 1.21	90.50 ± 1.16



(a) Payload (b) Protocol time (c) Encrypted chunks

Figure 1: Cost per round with ResNet-18-GN on CIFAR-10 (seed 123, rounds 1–2, setup spread over ten rounds). (a) Payload of five uploads and five deliveries; the dashed line counts only plaintext values and ciphertexts of the baselines. (b) Protocol phases without local training or network; colors as in (a). (c) Encrypted chunks per client, in the measured aggregate, for independent uniform masks, and at the bound of Proposition 2.

The gap to our FAS and ParaAegis versions. DiSHE is 20.72–49.51 points above our FAS adaptation and 4.96–13.40 points above our ParaAegis implementation. Both clip the unencrypted part of every update to norm 1 and add Gaussian noise; in all 900 logged client-rounds this part was clipped, by a coefficient between 0.020 and 0.073 (Table 7 in Appendix H), and with five equal clients the root-mean-square norm of the aggregate noise over the roughly 90% unencrypted coordinates, $\sigma\sqrt{0.9d/5}$, is about 4.3 on CIFAR-10 and 21.5 elsewhere, against at most 1 for the clipped aggregate. ParaAegis keeps the largest-magnitude coordinates unperturbed while FAS picks them at random, consistent with its higher accuracy but not a controlled comparison. No method here has a verified end-to-end DP guarantee, so privacy levels are not matched, and the gaps describe our clipping-and-noise settings rather than the published methods.

6.3 Communication and Computation

Accounting. A round costs the mean of rounds 1 and 2, with round 0 as warm-up, plus one tenth of any one-time setup, as for a ten-round run. Payload counts the application bytes of five uploads and five deliveries, including metadata, mask messages, and amortized setup, with 4 bytes per padded plaintext value. Protocol time sums the instrumented phases other than local training and excludes network transfer, so it is not an end-to-end latency; since the probe always has 512 examples, the costs of DiSHE do not depend on the size of client datasets (Appendix F).

Results. At a 10% ratio, DiSHE sends 965 MB per round, 45–56% less than the baselines, and has the lowest protocol time, 1.51 s against 1.82–15.54 s (Figure 1; Table 5 in Appendix F). It runs no consensus and sends no mask messages, whereas MaskCrypt spends 0.51 s and 44.7 MB per round on consensus and ParaAegis 13.40 s and 268 MB; FedML-HE instead pays 20.9 s and 4.6 GB once for its sensitivity maps.

Table 2: Reconstruction quality of DLG and IG against LeNet-5 on Fashion-MNIST at $\alpha = 0.05$, at rounds 0, 4, and 8. Arrows point toward stronger protection (a worse reconstruction). Every attacker knows the true label except in the second row. Values average three targets and three seeds and, for DiSHE, all five clients.

	DLG						IG					
	SSIM ↓			LPIPS ↑			SSIM ↓			LPIPS ↑		
Method	0	4	8	0	4	8	0	4	8	0	4	8
DiSHE	.372	.134	.093	.193	.475	.507	.226	.281	.243	.362	.390	.393
DiSHE, label unknown	.416	.065	.086	.188	.522	.524	.237	.232	.180	.365	.393	.439
FAS	.107	.275	.063	.493	.552	.530	.045	.191	.171	.569	.482	.482
FedML-HE	.295	.079	.044	.240	.532	.537	.136	.179	.146	.438	.477	.476
MaskCrypt	.328	.152	.099	.207	.504	.524	.209	.153	.145	.382	.511	.524
ParaAegis	.079	.270	.075	.516	.538	.563	.031	.109	.105	.569	.521	.527

The payload breakdown shows where the gap comes from. Every baseline message carries about 89 MB of metadata, the same for all four baselines and all ratios, so it reflects their shared message format; the flags and scores of DiSHE take 0.4 MB per round. Counting only plaintext values and ciphertexts, DiSHE sends about 12% more than the baselines (965 against 862 MB), because its broadcast encrypts every chunk that some client encrypted. The payload advantage therefore comes from the absence of per-coordinate metadata and mask messages, not from fewer ciphertexts; a more compact baseline format would narrow it, or reverse it against FAS and MaskCrypt.

The union of the client masks stays far below its worst case (Figure 1c): the aggregate encrypts 8.8, 16.2, 34.8, and 81.5% of the chunks at requested ratios of 5, 10, 25, and 75%. At 10% this is 444 chunks on average, against the bound of 1,367 (Proposition 2) and the 1,121 expected for independent uniform masks, so the CGD rankings of different clients largely agree. With nothing encrypted, DiSHE is slower than MaskCrypt and ParaAegis (0.47 s against 0.14–0.15 s), a gap that includes the CGD probe (Appendix F gives the other endpoint).

6.4 Gradient inversion and label attacks

Protocol. We attack LeNet-5 on Fashion-MNIST at $\alpha = 0.05$ ($K = 12$ of 243 chunks, four of them the head) at rounds 0, 4, and 8 of the ten-round runs. We deliberately evaluate an attacker-favorable gradient-access setting. The target gradient is computed at the true round-entry model for an individual example, or a batch for label counting. The attacker receives its visible coordinates under the encryption mask, with the clipping and noise of our implementations applied for FAS and ParaAegis. The attacker computes dummy gradients at its own estimate of the model (for DiSHE, Equation (7)) and matches the visible coordinates only. We run DLG (Zhu et al., 2019) and IG (Geiping et al., 2020) with the true label supplied to every method and, for DiSHE only, with the label optimized jointly, plus the analytic label attacks iDLG (Zhao et al., 2020) and iLRG (Ma et al., 2023). Table 2 reports SSIM (Wang et al., 2004) and LPIPS (Zhang et al., 2018), and Appendix J the settings, MSE, PSNR, and the most exposed client. This setting gives the adversary direct access to the unencrypted coordinates of target gradients, testing protection without relying on the additional difficulty introduced by accumulating updates over many local training steps.

Label attacks. Because every client of DiSHE encrypts the whole head, neither analytic attack obtains the gradients it needs: iDLG recovers no label in 135 attempts and iLRG cannot be run. Against the baselines, iDLG succeeds in 27 of 27 attempts for FedML-HE and MaskCrypt and in 9 of 27 for FAS and ParaAegis, and iLRG recovers batch labels with an F1 score of 0.89–0.90 for all of them except FedML-HE, whose mask also hides the final-layer bias. Optimization still finds labels: the jointly optimized labels are correct in 56–100% of the DiSHE cases, depending on round and attack.

Image reconstruction. With the true label supplied to every method, DiSHE does not make images harder to reconstruct than the baselines (Table 2). Its LPIPS is the lowest of the five methods at every round under both attacks, and under IG its SSIM is also the highest. Under DLG its SSIM is close to that of MaskCrypt: both leave recognizable images at round 0 (0.37 and 0.33), and at round 4 DiSHE

is below MaskCrypt, FAS, and ParaAegis but above FedML-HE. FAS and ParaAegis resist best on most metrics, consistent with their noise and their loss of accuracy. Without the label, DLG against DiSHE degrades at rounds 4 and 8 but not at round 0; this setting has no baseline counterpart. We thus find no evidence that CGD selection protects images better than the masks of the baselines at this budget, where the head takes 4 of the 12 encrypted chunks and selection works on 256-value chunks rather than on coordinates; neither factor was tested separately.

7 Limitations

DiSHE assumes an honest-but-curious server, one shared key, no collusion, and full participation; tolerating collusion would require multiparty CKKS (Mouchet et al., 2021), and our simulation separates keys only at the level of software interfaces. The gradient-access evaluation covers LeNet-5 at a 5% encryption ratio, with three targets and three seeds; unknown-label reconstruction is evaluated only for DiSHE. Flags and scores, including those of encrypted chunks such as the head, travel in the clear and reveal data-dependent statistics (Remark 5); DiSHE$_0$ would not need the scores. We evaluated only the $\gamma = 1$ end of the family and ran no ablations of selection, allocation, head policy, or aggregation, so we cannot say which choice limits the image protection. The accuracy results use five clients, ten rounds, three seeds, and partitions that may overlap and are not paired across methods; the cost results come from one seed and two post-warm-up rounds and depend on our serializers; and the baselines are reimplementations.

8 Conclusion

DiSHE shows that selective HE in FL does not require clients to agree on a mask: a ciphertext-aligned chunk grid lets each client choose its own chunks, guided by a round-start dispersion score. Because encryption leaves the aggregated values unchanged, the aggregation rule alone determines utility in exact arithmetic; we analyze it as a family from FedAvg to inverse-dispersion weighting with a cost independent of the member. Our measurements show that independently chosen client masks can overlap strongly, keeping the encrypted broadcast compact. At a 10% client budget, the aggregate encrypts 16.2% of chunks, and DiSHE achieves the lowest measured protocol time of the five implementations. In the LeNet-5 attack evaluation, its encrypted head blocks both analytic label attacks in every attempt. Multiparty keys, partial participation, and intermediate values of γ are natural next steps.

AI use statement

In this work, we used generative AI tools (large language models) as coding assistants to speed up the implementation of our method and some baselines, check that the manuscript matches the implementation, help write and polish the text, and verify and format the references. We did not use generative AI to generate or alter experimental data. We have reviewed all AI-assisted work: we checked the derivations, checked the reported numbers against the run outputs, verified the references against their original sources, and reviewed and tested the AI-assisted code before running the experiments reported here. We take full responsibility for the content of this paper, including text, claims, and artifacts produced with the aid of generative AI.

Ethics statement

This work proposes a defense that reduces what an aggregation server can learn about clients' training data, and it uses only public benchmark datasets. DiSHE provides no formal privacy guarantee, and its protection holds only under the stated honest-but-curious threat model; deployments facing malicious servers, collusion, or key compromise need additional mechanisms.

Reproducibility statement

Section 4 and Algorithms 1 to 3 (Appendix C) specify the protocol, including tie-breaking and rounding, and Appendix B proves every formal claim under the stated assumptions. Appendix E gives the chunk map and budgets, Appendix F the CKKS parameters and the cost accounting,

Appendix G the runs and data partitioning, Appendix H the baseline implementations and their deviations from the original papers, Appendix I the accuracy statistics, and Appendix J the attack settings. Anonymized code for training, encryption, cost measurement, and the attacks is available at https://github.com/anonymous-researcher2/DiSHE.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 308–318, 2016.

Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. Homomorphic encryption standard. In Kristin Lauter, Wei Dai, and Kim Laine (eds.), *Protecting Privacy through Homomorphic Encryption*, pp. 31–62. Springer, Cham, 2021. doi: 10.1007/978-3-030-77287-1_2.

Mislav Balunović, Dimitar I. Dimitrov, Robin Staab, and Martin Vechev. Bayesian framework for gradient leakage. In *International Conference on Learning Representations (ICLR)*, 2022.

Ayoub Benaissa, Bilal Retiat, Bogdan Cebere, and Alaa Eddine Belfedhal. TenSEAL: A library for encrypted tensor operations using homomorphic encryption. In *ICLR 2021 Workshop on Distributed and Private Machine Learning (DPML)*, 2021.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 1175–1191, 2017.

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology – ASIACRYPT 2017*, volume 10624 of *Lecture Notes in Computer Science*, pp. 409–437. Springer, 2017.

William G. Cochran. The combination of estimates from different experiments. *Biometrics*, 10(1): 101–129, 1954.

William G. Cochran and Sarah Porter Carroll. A sampling investigation of the efficiency of weighting inversely as the estimated variance. *Biometrics*, 9(4):447–459, 1953.

Dimitar I. Dimitrov, Mislav Balunović, Nikola Konstantinov, and Martin Vechev. Data leakage in federated averaging. *Transactions on Machine Learning Research*, 2022.

Liam Fowl, Jonas Geiping, Wojtek Czaja, Micah Goldblum, and Tom Goldstein. Robbing the fed: Directly obtaining private data in federated learning with modified models. In *International Conference on Learning Representations (ICLR)*, 2022.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients – how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 16937–16947, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-IID data quagmire of decentralized machine learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4387–4398, 2020.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Chenghao Hu and Baochun Li. MaskCrypt: Federated learning with selective homomorphic encryption. *IEEE Transactions on Dependable and Secure Computing*, 22(1):221–233, 2025. doi: 10.1109/TDSC.2024.3392424.

Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 7232–7241, 2021.

Weizhao Jin, Yuhang Yao, Shanshan Han, Jiajun Gu, Carlee Joe-Wong, Srivatsan Ravi, Salman Avestimehr, and Chaoyang He. FedML-HE: An efficient homomorphic-encryption-based privacy-preserving federated learning system. *arXiv preprint arXiv:2303.10837*, 2023.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14 (1–2):1–210, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Leslie Kish. *Survey Sampling*. John Wiley & Sons, New York, 1965.

Abdulkadir Korkmaz and Praveen Rao. A selective homomorphic encryption approach for faster privacy-preserving federated learning. *arXiv preprint arXiv:2501.12911*, 2025. Conference version published as "Fast and Secure Selective Homomorphic Encryption for Federated Learning," IEEE CCNC, 2026. DOI: 10.1109/CCNC65079.2026.11366371.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Kailang Ma, Yu Sun, Jian Cui, Dawei Li, Zhenyu Guan, and Jianwei Liu. Instance-wise batch label restoration via gradients in federated learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, 2017.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations (ICLR)*, 2018.

Christian Mouchet, Juan Troncoso-Pastoriza, Jean-Philippe Bossuat, and Jean-Pierre Hubaux. Multiparty homomorphic encryption from ring-learning-with-errors. *Proceedings on Privacy Enhancing Technologies*, 2021(4):291–311, 2021. doi: 10.2478/popets-2021-0071.

Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, pp. 739–753, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2018.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017.

Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security (AISec)*, pp. 1–11, 2019.

Aidmar Wainakh, Fabrizio Ventola, Till Müßig, Jens Keim, Carlos Garcia Cordero, Ephraim Zimmer, Tim Grube, Kristian Kersting, and Max Mühlhäuser. User-level label leakage from gradients in federated learning. *Proceedings on Privacy Enhancing Technologies*, 2022(2):227–244, 2022.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

Zihou Wu, Yuecheng Li, Tianchi Liao, Jian Lou, and Chuan Chen. ParaAegis: Parallel protection for flexible privacy-preserved federated learning. *arXiv preprint arXiv:2509.13739*, 2025. Conference version published at IEEE ICASSP, 2026. DOI: 10.1109/ICASSP55912.2026.11463612.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Dong Yin, Ashwin Pananjady, Max Lam, Dimitris Papailiopoulos, Kannan Ramchandran, and Peter Bartlett. Gradient diversity: a key ingredient for scalable distributed learning. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1998–2007, 2018.

Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M. Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via GradInversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16337–16346, 2021.

Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC)*, pp. 493–506, 2020.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. iDLG: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

A NOTATION

Symbol	Meaning
N, i; T, t; E	number of clients, client index; rounds, round index (from 0); local epochs (many Adam steps each)
$\mathcal{D}_i$, n_i, a_i	local dataset, its size, FedAvg weight $n_i / \sum_r n_r$
θ^t, Δ_i^t	global model at the start of round t; update $\theta_i^{t,\mathrm{loc}} - \theta^t$ of client i
l, $\mathcal{C}_l$, $\mathcal{C}$, C	logical layer, its chunks, all chunks, number of chunks
b, S, d_c, pad_b	logical chunk length, CKKS slots per ciphertext, real values in chunk c, zero-padding to length b
$\mathcal{B}_i^t$, m_i	probe of client i in round t, its size $\min(512, n_i)$
$g_{i,c,j}$, $G_{i,c}^t$, $Q_{i,c}^t$	per-example gradient on chunk c at θ^t; its sum and sum of squared norms over the probe
$\widehat{D}_{i,c}^t$	chunk gradient dispersion (Equation (1))
α, K, K_{free}, α_{eff}	requested encryption ratio, encrypted chunks per client, free budget, effective ratio (Equation (2))
$\mathcal{M}$, $\mathcal{F}_l$, $\omega_{i,l}$, $q_{i,l}$	head chunks, free chunks of layer l, layer weight N_{eff}, layer quota
$\mathcal{E}_i^t$, $\mathcal{U}^t$	encrypted set of client i; union $\bigcup_i \mathcal{E}_i^t$
$\mathcal{H}_c^t$, $\mathcal{V}_c^t$	clients that sent chunk c encrypted, in plaintext
ε_c^t, $\pi_{i,c}^t$, $w_{i,c}^t$	median offset, precision, inverse-dispersion weight (Equation (5))
γ, $\lambda_{i,c}^t(\gamma)$	family index ($\gamma = 1$ evaluated), aggregation weight $(1-\gamma)a_i + \gamma w_{i,c}^t$
U^t, $\hat{\theta}^t$	broadcast aggregate (Equation (6)), server's visible-update estimate (Equation (7))
κ_c, r_c	$\sum_i \lvert w_{i,c} - a_i \rvert$, $\max_i \lVert \Delta_{i,c} - U_c(0) \rVert_2$ (Proposition 1)

B PROOFS

The following identity underlies the reading of CGD given in Section 5.

Proposition 4 (Probe leave-one-out identity)**.** *Fix client i, round t, and chunk c with $m = m_i \geq 2$; write $g_j = g_{i,c,j}$ and $\bar{g} = \bar{g}_{i,c}$. For a step size $\eta > 0$, let $u_c = -\eta\, \bar{g}$ be a hypothetical gradient step on the probe mean at θ^t, and $u_c^{(-j)} = -\eta \frac{1}{m-1} \sum_{k \neq j} g_k$ the same step without example j. Then $u_c - u_c^{(-j)} = -\frac{\eta}{m-1}(g_j - \bar{g})$ and*

$$\frac{1}{m\, d_c} \sum_{j \in \mathcal{B}_i^t} \left\| u_c - u_c^{(-j)} \right\|_2^2 = \frac{\eta^2}{m(m-1)}\, \widehat{D}_{i,c}^t. \tag{10}$$

Proof of Proposition 4. The mean over the probe without example j is $\bar{g}^{(-j)} = (m\bar{g} - g_j)/(m-1)$, so

$$\bar{g} - \bar{g}^{(-j)} = \frac{(m-1)\bar{g} - m\bar{g} + g_j}{m-1} = \frac{g_j - \bar{g}}{m-1},$$

and multiplying by $-\eta$ gives $u_c - u_c^{(-j)} = -\eta(g_j - \bar{g})/(m-1)$. Hence $\sum_j \left\| u_c - u_c^{(-j)} \right\|_2^2 = \frac{\eta^2}{(m-1)^2} \sum_j \|g_j - \bar{g}\|_2^2 = \frac{\eta^2}{(m-1)^2}(m-1) d_c \widehat{D}_{i,c}^t$ by Equation (1), where the clamp is inactive because $\sum_j \|g_j - \bar{g}\|^2 = Q_{i,c}^t - \left\|G_{i,c}^t\right\|^2 / m \geq 0$ in exact arithmetic. Dividing by $m\, d_c$ gives Equation (10). □

The allocation rule of Section 4.3 relies on the following properties of the effective count.

Lemma 1 (Effective count)**.** *For $\boldsymbol{x} \in \mathbb{R}_{\geq 0}^k \setminus \{\mathbf{0}\}$: (i) $N_{\mathrm{eff}}(s\boldsymbol{x}) = N_{\mathrm{eff}}(\boldsymbol{x})$ for all $s > 0$; (ii) $1 \leq N_{\mathrm{eff}}(\boldsymbol{x}) \leq \|\boldsymbol{x}\|_0$, where $\|\boldsymbol{x}\|_0$ is the number of nonzero entries; (iii) the lower bound is attained if and only if exactly one entry is nonzero, and the upper bound if and only if all nonzero entries are equal.*

Proof. (i) $N_{\mathrm{eff}}(s\boldsymbol{x}) = s^2(\sum_c x_c)^2 / (s^2 \sum_c x_c^2) = N_{\mathrm{eff}}(\boldsymbol{x})$. (ii)–(iii) Let A be the support of $\boldsymbol{x}$. Because all entries are nonnegative, $(\sum_c x_c)^2 = \sum_c x_c^2 + \sum_{c \neq c'} x_c x_{c'} \geq \sum_c x_c^2$, with equality

if and only if every cross term vanishes, that is, if and only if $|A| = 1$. By the Cauchy–Schwarz inequality on A, $(\sum_{c\in A} x_c)^2 \le |A| \sum_{c\in A} x_c^2$, with equality if and only if x_c is constant on A. Dividing both inequalities by $\sum_c x_c^2 > 0$ completes the proof. □

Proof of Remark 1. Consider two runs that differ only in α or in the rule that chooses the encrypted sets, and suppose they share θ^t. Local training starts from θ^t with the same seeds and data, so both runs produce the same updates Δ_i^t; the probe depends only on the seeds, the round and the client, so both compute the same scores, hence the same weights $\lambda_{i,c}^t(\gamma)$ by Equation (5). By the homomorphic properties of the scheme in exact arithmetic, decrypting Equation (6) gives, on the real coordinates of chunk c,

$$\sum_{i\in\mathcal{H}_c^t} \lambda_{i,c}^t \Delta_{i,c}^t + \sum_{i\in\mathcal{V}_c^t} \lambda_{i,c}^t \Delta_{i,c}^t = \sum_i \lambda_{i,c}^t \Delta_{i,c}^t,$$

which does not depend on how the clients are split between $\mathcal{H}_c^t$ and $\mathcal{V}_c^t$; the padding is zero and discarded. Both runs therefore set the same θ^{t+1}, and induction from the common θ^0 proves the first claim. For the second, aggregating updates with data-size weights gives $\theta^t + \sum_i a_i \Delta_i^t$, and aggregating local models gives $\sum_i a_i \theta_i^{t,\mathrm{loc}} = \sum_i a_i(\theta^t + \Delta_i^t) = \theta^t + \sum_i a_i \Delta_i^t$ because $\sum_i a_i = 1$. Since $\lambda_{i,c}(0) = a_i$, this is the round-t result of DiSHE$_0$ whenever the local updates coincide, and encrypted coordinates decrypt to the values they encrypt, so induction again gives the trajectory of DiSHE$_0$. □

Part (c) of Proposition 1 uses the following one-dimensional fact.

Lemma 2 (Linear interpolation of a function with Lipschitz derivative)**.** *Let $\varphi : [0,1] \to \mathbb{R}$ be differentiable with $|\varphi'(s) - \varphi'(u)| \le L'|s-u|$. Then $|\varphi(\gamma) - (1-\gamma)\varphi(0) - \gamma\varphi(1)| \le \frac{L'}{2}\gamma(1-\gamma)$ for all $\gamma \in [0,1]$.*

Proof. For either sign, $\psi_\pm(s) = \frac{L'}{2}s^2 \pm \varphi(s)$ has a nondecreasing derivative, since for $s < u$, $\psi'_\pm(u) - \psi'_\pm(s) = L'(u-s) \pm (\varphi'(u) - \varphi'(s)) \ge L'(u-s) - |\varphi'(u) - \varphi'(s)| \ge 0$; hence $\psi_\pm$ is convex and $\psi_\pm(\gamma) \le (1-\gamma)\psi_\pm(0) + \gamma\psi_\pm(1)$. For ψ_+ this reads $\frac{L'}{2}\gamma^2 + \varphi(\gamma) \le (1-\gamma)\varphi(0) + \gamma\big(\frac{L'}{2} + \varphi(1)\big)$, that is, $\varphi(\gamma) - (1-\gamma)\varphi(0) - \gamma\varphi(1) \le \frac{L'}{2}(\gamma - \gamma^2)$; for ψ_- it gives the same bound on the negative of the left-hand side. □

Proof of Proposition 1. (a) Since $n_i > 0$ and $\widehat{D}_{i,c} + \varepsilon_c \ge 10^{-12} > 0$, every $\pi_{i,c}$ is positive, so $w_{\cdot,c}$ is a probability vector with entries in $(0,1)$, as is a. Their convex combination $\lambda_{\cdot,c}(\gamma) = (1-\gamma)a + \gamma w_{\cdot,c}$ is again one, and $w_{i,c} \in (0,1)$ gives $(1-\gamma)a_i \le \lambda_{i,c}(\gamma) \le (1-\gamma)a_i + \gamma$. For $\gamma = 0$, $\lambda_{\cdot,c} = a$; if $\widehat{D}_{i,c} = x$ for all i, then $\pi_{i,c} = n_i/(x + \varepsilon_c) \propto n_i$, so $w_{\cdot,c} = a$ and $\lambda_{\cdot,c}(\gamma) = a$ for every γ. (b) Chunkwise, $U_c(\gamma) = \sum_i \big((1-\gamma)a_i + \gamma w_{i,c}\big)\Delta_{i,c} = (1-\gamma)U_c(0) + \gamma U_c(1)$, and adding θ gives $x_\gamma = (1-\gamma)x_0 + \gamma x_1$. Both weight vectors sum to one, so

$$U_c(1) - U_c(0) = \sum_i (w_{i,c} - a_i)\,\Delta_{i,c} = \sum_i (w_{i,c} - a_i)\big(\Delta_{i,c} - U_c(0)\big),$$

and the triangle inequality gives $\|U_c(1) - U_c(0)\|_2 \le \kappa_c r_c$, with $\kappa_c \le \sum_i w_{i,c} + \sum_i a_i = 2$. Summing the squares over chunks gives $\|x_1 - x_0\|^2 \le \sum_c \kappa_c^2 r_c^2$. (c) Let $\varphi(s) = F(x_s)$ with $x_s = x_0 + s(x_1 - x_0)$. Then $\varphi'(s) = \langle \nabla F(x_s), x_1 - x_0\rangle$, so $|\varphi'(s) - \varphi'(u)| \le \|\nabla F(x_s) - \nabla F(x_u)\|\,\|x_1 - x_0\| \le L|s-u|\,\|x_1 - x_0\|^2$, and Lemma 2 with $L' = L\,\|x_1 - x_0\|^2$ gives Equation (8); since $\gamma(1-\gamma) \le \frac{1}{4}$, the bound is at most $\frac{L}{8}\,\|x_1 - x_0\|^2$. □

Remark 3 (Concentration of inverse-dispersion weights). Take five clients of equal size and scores $(0,0,0,1,1)$ on some chunk. The median is 0, so $\varepsilon_c = 10^{-12}$, and $w_{\cdot,c} \approx (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 3.3\times10^{-13}, 3.3\times10^{-13})$. If only the last two clients change the chunk, by u each, FedAvg aggregates $0.4u$ and $\gamma = 1$ aggregates about $6.7 \times 10^{-13}u$: the two high-dispersion clients are silenced on that chunk. The weights remain positive and normalized, so this does not contradict Proposition 1; it shows that $\gamma = 1$ need not preserve the contribution of heterogeneous clients, which part (a) guarantees for $\gamma < 1$ at a level $(1-\gamma)a_i$. How often near-zero medians occur in practice is an empirical question that we do not answer here.

Remark 4 (Equal endpoints do not imply a good midpoint). For $F(x) = (x^2 - 1)^2$, $x_0 = -1$, and $x_1 = 1$, both endpoints have $F = 0$ while $F(x_{1/2}) = 1$. This is consistent with Equation (8): F'' lies in $[-4, 8]$ on $[-1, 1]$, so $L = 8$ and the bound at $\gamma = \frac{1}{2}$ is $\frac{L}{8} \|x_1 - x_0\|^2 = 4$. Proposition 1(c) therefore yields comparable intermediate values only when $L \|x_1 - x_0\|^2$ is small relative to the endpoint values.

The following proposition, summarized in Section 5, states the multi-round result.

Proposition 5 (Conditional stability across rounds)**.** *Let Φ_γ^t be the round-t map $\theta \mapsto \theta + \sum_i \lambda_i^t(\gamma;\theta)\Delta_i^t(\theta)$ of DiSHE$_\gamma$ with all seeds fixed. Then $\Phi_\gamma^t = (1-\gamma)\Phi_0^t + \gamma\Phi_1^t$, and if each Φ_γ^t is ρ_t-Lipschitz on a set containing the compared iterates, the trajectories from a common θ^0 satisfy, for $e \in \{0, 1\}$,*

$$\left\|\theta_\gamma^T - \theta_e^T\right\| \;\le\; |\gamma - e| \sum_{t=0}^{T-1} \Big(\prod_{s=t+1}^{T-1} \rho_s \Big)\, \delta_e^t, \qquad \delta_e^t = \left\|\Phi_1^t(\theta_e^t) - \Phi_0^t(\theta_e^t)\right\|. \tag{11}$$

Proof of Proposition 5. Since $\Delta_i^t(\theta)$ and $w^t(\theta)$ do not depend on γ, $\Phi_\gamma^t(\theta) = \theta + \sum_i \big((1-\gamma)a_i + \gamma w_i^t(\theta)\big)\Delta_i^t(\theta) = (1-\gamma)\Phi_0^t(\theta) + \gamma\Phi_1^t(\theta)$, with weights applied chunkwise. Consequently $\Phi_\gamma^t - \Phi_e^t = (\gamma - e)(\Phi_1^t - \Phi_0^t)$ for $e \in \{0, 1\}$. Let $\epsilon_t = \left\|\theta_\gamma^t - \theta_e^t\right\|$, so $\epsilon_0 = 0$. Then

$$\epsilon_{t+1} \le \left\|\Phi_\gamma^t(\theta_\gamma^t) - \Phi_\gamma^t(\theta_e^t)\right\| + \left\|\Phi_\gamma^t(\theta_e^t) - \Phi_e^t(\theta_e^t)\right\| \le \rho_t\, \epsilon_t + |\gamma - e|\, \delta_e^t,$$

and unrolling the recursion gives Equation (11). □

Remark 5 (Scores can leak label statistics). Consider a chunk consisting of one bias of a binary classifier whose predicted probability of class 1 is p for every probe example. The per-example bias gradient is $p - y_j$, so if a fraction q of the m probe labels equals 1, the sample variance, and hence the CGD of that chunk, is $\frac{m}{m-1} q(1-q)$, which reveals the label balance up to $q \leftrightarrow 1 - q$ even though the update of the bias may be encrypted. Real chunks mix many parameters, so this is not an attack on ResNet-18 scores; it shows that encrypting a chunk does not hide its score, which DiSHE sends in the clear for every chunk.

Proof of Proposition 2. By Equation (6), U_c^t is a ciphertext if and only if $\mathcal{H}_c^t \neq \emptyset$, that is, if and only if $c \in \bigcup_i \mathcal{E}_i^t$. For $\alpha = 0$ all sets are empty. For $\alpha > 0$, the union contains $\mathcal{E}_1^t$, so $|\mathcal{U}^t| \ge K$. Since every $\mathcal{E}_i^t$ contains $\mathcal{M}$, $|\mathcal{U}^t| = |\mathcal{M} \cup \bigcup_i (\mathcal{E}_i^t \setminus \mathcal{M})| \le |\mathcal{M}| + N(K - |\mathcal{M}|)$, and $|\mathcal{U}^t| \le C$ trivially; no independence between the masks is used. The stateless server broadcasts only U^t, so the bound applies to each round separately. If the server instead stored a global state and added U^t to it, a chunk would become a ciphertext as soon as any aggregate for it was one and would remain one, because adding a plaintext or a ciphertext to a ciphertext yields a ciphertext; after round t the encrypted chunks would be $\bigcup_{s \le t} \mathcal{U}^s$, which can only grow with t. □

Proof of Proposition 3. Clients apply the decrypted aggregate, so $\theta_c^{t+1} = \theta_c^t + \sum_{i \in \mathcal{V}_c^t} \lambda_{i,c}^t \Delta_{i,c}^t + \sum_{i \in \mathcal{H}_c^t} \lambda_{i,c}^t \Delta_{i,c}^t$. Subtracting Equation (7) gives $\theta_c^{t+1} - \hat{\theta}_c^{t+1} = \theta_c^t - \hat{\theta}_c^t + \sum_{i \in \mathcal{H}_c^t} \lambda_{i,c}^t \Delta_{i,c}^t$, and induction from $\hat{\theta}^0 = \theta^0$ yields the identity. If $\mathcal{H}_c^s = \emptyset$ for all $s < t$, the sum is empty. Unrolling Equation (7) shows $\hat{\theta}_c^t = \theta_c^0 + \sum_{s<t} \sum_{i \in \mathcal{V}_c^s} \lambda_{i,c}^s \Delta_{i,c}^s$, which contains no term indexed by any $\mathcal{H}_c^s$. □

C Algorithms

A pass with positive total weight either places all remaining units or saturates at least one layer, and a zero-weight pass places one unit in every active layer (all of which have spare capacity), so the loop terminates after at most (number of layers) + K_{free} passes, with exactly K_{free} units placed and every capacity respected; a single zero-weight layer that absorbs all remaining units takes one pass per unit. The capacity condition holds because $K \le C$ for $\alpha \le 1$. A layer whose free chunks all have zero dispersion can still receive chunks through the capacity fallback.

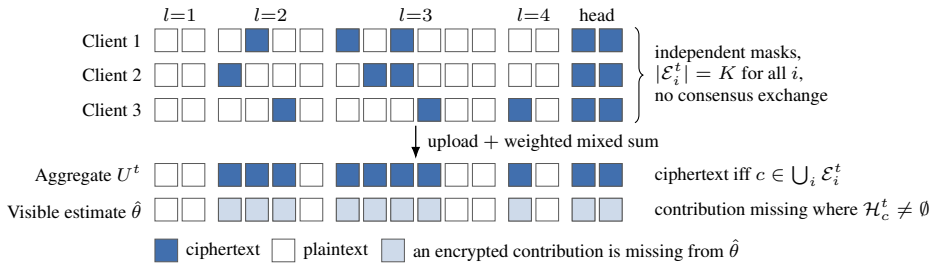


Figure 2: One round of DiSHE on an illustrative model with 16 chunks. Each client encrypts $K = 5$ chunks (two head chunks, three chosen by CGD) without a consensus exchange; a chunk occupies the same slots for every client, and an aggregate chunk is a ciphertext exactly when some client encrypted it.

Algorithm 1 DiSHE: client i in round t

Require: global model θ^t, data $\mathcal{D}_i$, ratio α, public chunk map $(\mathcal{C}_l)_l$, head $\mathcal{M}$, CKKS key

load $\theta \leftarrow \theta^t$; reset the Adam state

draw a probe $\mathcal{B} \subseteq \mathcal{D}_i$ of $m = \min(512, n_i)$ examples uniformly without replacement ▷ seeded by (seed, t, i)

$G_c \leftarrow \mathbf{0}$, $Q_c \leftarrow 0$ for all $c \in \mathcal{C}$

for each micro-block of 256 probe examples **do** ▷ evaluation mode, θ^t frozen

  compute per-example gradients $g_{c,j}$ at θ^t (vectorized); $G_c \leftarrow G_c + \sum_j g_{c,j}$, $Q_c \leftarrow Q_c + \sum_j \|g_{c,j}\|_2^2$

end for

$\widehat{D}_c \leftarrow \max\{0, (Q_c - \|G_c\|_2^2/m)/((m-1)d_c)\}$ for all c (and $\widehat{D}_c \leftarrow 0$ if $m = 1$) ▷ Equation (1)

run E epochs of Adam (batch 128, stored order) on $\mathcal{D}_i$; $\Delta \leftarrow \theta - \theta^t$

if $\alpha = 0$ **then** $\mathcal{E} \leftarrow \emptyset$

else

  $K \leftarrow \max(\mathrm{rnd}(\alpha C), |\mathcal{M}|)$; $K_{\text{free}} \leftarrow K - |\mathcal{M}|$ ▷ ties to even

  $\omega_l \leftarrow N_{\text{eff}}\big((\widehat{D}_c)_{c \in \mathcal{F}_l}\big)$; $u_l \leftarrow |\mathcal{F}_l|$ for every layer l

  $(q_l)_l \leftarrow \textsc{Quota}\big((\omega_l)_l, (u_l)_l, K_{\text{free}}\big)$ ▷ Algorithm 3

  $\mathcal{E} \leftarrow \mathcal{M} \cup \bigcup_l \{$the q_l chunks of $\mathcal{F}_l$ with largest $\widehat{D}_c$, ties by global index$\}$

end if

for each chunk c in the public order **do**

  $V_c \leftarrow \mathsf{Enc}(\mathrm{pad}_b(\Delta_c))$ if $c \in \mathcal{E}$, else $V_c \leftarrow \mathrm{pad}_b(\Delta_c)$ ▷ float32, b values

end for

send $(V_c, \mathbb{1}[c \in \mathcal{E}],\ \widehat{D}_c)_{c \in \mathcal{C}}$ ▷ flags Boolean, scores float32

receive U^t and its flags; decrypt ciphertext chunks, strip padding; $\theta^{t+1} \leftarrow \theta^t + U^t$

D Implementation Details

Model state and chunk map. The model state must consist of trainable parameters only; buffers, frozen parameters, and aliased tensors are rejected, and no batch-normalization statistics exist. The registration order of the parameters determines the public map and the global chunk indices used for tie-breaking.

Probe and CGD. The probe is resampled every round, with a seed derived from the experiment seed, the zero-based round index, the client index, and a CGD-specific domain. Gradients are computed in evaluation mode at the frozen incoming model, before local training. Per-example gradients are obtained with vectorized automatic differentiation for one micro-block of 256 examples at a time (two micro-blocks for a full probe), so the transient memory is roughly the micro-block size times the number of parameters; for ResNet-18-GN this is 11.4 GB (10.7 GiB) of float32 per-example gradients, before activations and framework overhead. Only a model-sized first-moment sum and one squared-norm sum per chunk persist across micro-blocks. First-moment sums are float32, squared-norm sums and the final arithmetic of Equation (1) are float32, small negative cancellation residuals are clamped

Algorithm 2 DiSHE$_\gamma$: server in round t (all experiments use $\gamma = 1$)

Require: payloads $(V_{i,c}, f_{i,c}, \widehat{D}_{i,c})$ of all clients, sizes n_i, $\gamma \in [0, 1]$
for each chunk c **do**
  $\varepsilon_c \leftarrow \max(\mathrm{median}_i\, \widehat{D}_{i,c}, 10^{-12})$; $\pi_{i,c} \leftarrow n_i/(\widehat{D}_{i,c} + \varepsilon_c)$; $w_{i,c} \leftarrow \pi_{i,c}/\sum_r \pi_{r,c}$
  $\lambda_{i,c} \leftarrow (1-\gamma)\, a_i + \gamma\, w_{i,c}$ for all i
  $\mathcal{H}_c \leftarrow \{i : f_{i,c} = 1\}$; $\mathcal{V}_c \leftarrow \{i : f_{i,c} = 0\}$; $p_c \leftarrow \sum_{i \in \mathcal{V}_c} \lambda_{i,c} V_{i,c}$ ▷ plaintext subtotal
  if $\mathcal{H}_c = \emptyset$ **then** $U_c \leftarrow p_c$
  else $U_c \leftarrow \boxplus_{i \in \mathcal{H}_c} \lambda_{i,c} \boxdot V_{i,c}$; **if** $\mathcal{V}_c \neq \emptyset$ **then** $U_c \leftarrow U_c \boxplus p_c$ ▷ one rescale per term
  end if
  (what a curious server can track) $\hat{\theta}_c \leftarrow \hat{\theta}_c + p_c$ on the real coordinates ▷ Equation (7)
end for
broadcast $(U_c, \mathbb{1}[\mathcal{H}_c \neq \emptyset])_{c \in \mathcal{C}}$; keep no model state

Algorithm 3 QUOTA: capacity-aware largest-remainder allocation

Require: nonnegative weights $(\omega_l)_l$, capacities $(u_l)_l$ with $\sum_l u_l \geq K_{\mathrm{free}}$, total K_{free}
$q_l \leftarrow 0$ for all l; $A \leftarrow \{l : u_l > 0\}$
while $\sum_l q_l < K_{\mathrm{free}}$ **do**
  $R \leftarrow K_{\mathrm{free}} - \sum_l q_l$; $W \leftarrow \sum_{l \in A} \omega_l$
  if $W = 0$ **then** ▷ all remaining weights vanish
    for $l \in A$ in increasing layer index **do** ▷ one unit per active layer per pass
      if $R > 0$ **then** $q_l \leftarrow q_l + 1$; $R \leftarrow R - 1$
      end if
    end for
  else
    $q_l^* \leftarrow R\,\omega_l / W$ for $l \in A$; $q_l \leftarrow q_l + \min(\lfloor q_l^* \rfloor, u_l - q_l)$ for $l \in A$
    give the units still missing in this pass, at most one each, to the layers $l \in A$ with $q_l < u_l$ in decreasing order of $q_l^* - \lfloor q_l^* \rfloor$ (ties by layer index)
  end if
  $A \leftarrow \{l \in A : q_l < u_l\}$
end while
return $(q_l)_l$ ▷ $\sum_l q_l = K_{\mathrm{free}}$ and $q_l \leq u_l$

at zero, and the score is stored and sent as float32. Compilation of the per-example-gradient transform is requested with a fallback to eager execution, so the first measurement may include compilation time. Masks are selected after local training from these round-start scores; scores are not recomputed after training.

Local training. Each client resets Adam ($\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, no weight decay, no AMSGrad, no learning-rate schedule) whenever it loads the incoming model, and traverses its partition in stored order for $E = 5$ epochs with batch size 128 and learning rate 10^{-3}. Local training accuracy is evaluated after the last local epoch.

Allocation. Algorithm 3 distributes the remaining budget in proportion to the layer weights among unsaturated layers, assigns capped floor shares, then gives at most one extra unit per active layer in decreasing order of fractional remainder (ties by layer index), and repeats on the remaining active set when capacities bind; if all active weights vanish it deals one unit per active layer in increasing layer index. Positive rescaling of the scores within a layer leaves both its weight and its within-layer ranking unchanged, and a uniformly low-but-nonzero layer can receive the same weight as an equally sized, uniformly high one.

Payload, aggregation, and application. Plaintext chunks are sent as zero-padded float32 vectors of length b, flags as Booleans, and scores as float32 for every chunk. The server computes the median of Equation (5) separately for each chunk across clients and uses the full local dataset size n_i, not the probe size. Chunks encrypted by every client skip the addition of an all-zero plaintext subtotal. Because every client holds the same incoming model and key, the simulator runs one representative receive, decrypt, reconstruct, and apply operation per round.

E ARCHITECTURES, CHUNK MAPS, AND BUDGETS

ResNet-18-GN is the CIFAR variant of He et al. (2016): a 3×3 stem of stride one without initial pooling, basic blocks, convolutions without bias, and group normalization with 32 groups after every convolution (some modules keep names such as `bn1`, but no batch normalization or running statistics are used). Each convolution forms a logical layer with the group normalization that follows it; the two parameters of each downsampling shortcut form one logical layer; and the classifier $512 \to 10$ forms the last. With a one-channel input only the stem changes (704 instead of 1,856 parameters, still one chunk), so C and $|\mathcal{M}|$ are unchanged. Table 3 gives the chunk map. Densely packed into 4096-slot ciphertexts, the model would need 2,729 ciphertexts (2,728 with one input channel), whereas the all-encrypted endpoint of DiSHE uses one ciphertext per chunk, 2,746 in total. **LeNet-5** has a 5×5 convolution with 6 channels and padding 2, max pooling, a 5×5 convolution with 16 channels and no padding, max pooling, and fully connected layers $400 \to 120 \to 84 \to 10$ for 28×28 inputs; each layer with its bias is a logical layer. Because tail chunks are shorter than b, the number of encrypted real coordinates $\sum_{c \in \mathcal{E}_i} d_c$ can differ between clients even at fixed K.

Table 3: Chunk maps. Parameter counts d_l include biases and normalization parameters; chunks $= \lceil d_l/b \rceil$. Padding is the fraction of the bC slots that carry no parameter.

ResNet-18-GN, $b = 4096$, three-channel input Logical layer(s)	count	d_l each	chunks each	chunks
stem convolution + GN	1	1,856	1	1
stage 1 convolutions + GN	4	36,992	10	40
stage 2: first / shortcut	1 / 1	73,984 / 8,448	19 / 3	22
stage 2: other convolutions	3	147,712	37	111
stage 3: first / shortcut	1 / 1	295,424 / 33,280	73 / 9	82
stage 3: other convolutions	3	590,336	145	435
stage 4: first / shortcut	1 / 1	1,180,672 / 132,096	289 / 33	322
stage 4: other convolutions	3	2,360,320	577	1,731
classifier (head, $\mathcal{M}$)	1	5,130	2	2
total (padding 0.65%)	21	11,173,962 parameters		2,746

LeNet-5, $b = 256$, Fashion-MNIST ($28 \times 28 \times 1$) Logical layer	d_l	chunks
conv1 (with bias)	156	1
conv2 (with bias)	2,416	10
fc1	48,120	188
fc2	10,164	40
fc3 (head, $\mathcal{M}$)	850	4
total (padding 0.81%)	61,706	243

Table 4: Budgets for ResNet-18-GN ($C = 2{,}746$, $|\mathcal{M}| = 2$, $N = 5$): encrypted chunks K per client (Equation (2), ties to even), effective ratio $\alpha_{\text{eff}} = K/C$, the upper bound of Proposition 2 on the ciphertext chunks of the broadcast, the expected union for independent uniform masks, and the measured union in the cost runs (mean of rounds 1 and 2, seed 123).

α	K	α_{eff}	bound on $\lvert\mathcal{U}^t\rvert$	uniform reference	measured $\lvert\mathcal{U}^t\rvert$
0	0	0	0	0	0
0.05	137	0.0499	677	613.8	241
0.10	275	0.1001	1,367	1,121.1	444
0.25	686	0.2498	2,746	2,091.7	955
0.75	2,060	0.7502	2,746	2,743.3	2,238
1	2,746	1	2,746	2,746	2,746

Rounding with ties to even matters in one case: $0.25 \times 2{,}746 = 686.5$ gives $K = 686$, not 687. For LeNet-5 at $\alpha = 0.05$, the attack setting, $K = \text{rnd}(12.15) = 12$ ($\alpha_{\text{eff}} = 0.0494$), the bound of Proposition 2 is 44, and the expected union for independent uniform masks is 41.4. When $b < S$,

every chunk still occupies a whole ciphertext, so a round moves $N(K + |\mathcal{U}^t|)$ ciphertexts, against $2N\lceil d/S \rceil$ for densely packed full encryption; for this LeNet-5, $\lceil d/S \rceil = 16$, and the chunked protocol moves more ciphertexts than dense full encryption as soon as $|\mathcal{U}^t| \geq 21$.

F CKKS PARAMETERS, NUMERICAL EFFECTS, AND COST ACCOUNTING

Parameters. TenSEAL CKKS with polynomial modulus degree 8192 (4096 slots), coefficient-modulus bit sizes $[60, 30, 60]$ (150 bits), and global scale 2^{30}. Each fresh contribution undergoes one ciphertext–scalar multiplication and one rescaling, followed by additions; no ciphertext–ciphertext products, rotations, or bootstrapping are used, so these operations need neither relinearization nor Galois keys. Every encrypted chunk is one ciphertext.

Numerical effects. The identities of Section 5 assume exact arithmetic. In the simulation, CKKS approximation error, float32 model storage, and float32 scores introduce residuals, so the model trajectory is not bit-identical across encryption ratios, and the visible-update identity of Proposition 3 holds up to these residuals. We do not report a numerical error bound. Scores are computed from float32 first moments and float32 squared-norm sums; small negative cancellation residuals are clamped at zero.

Cost runs. Each cost run executes three rounds. Round 0 carries the one-time work (the chunk map of DiSHE, the sensitivity maps of FedML-HE, and the random mask of FAS) together with warm-up effects, such as the compilation of the per-example-gradient transform, which is requested with a fallback to eager execution. The recurring cost of a round is the mean of rounds 1 and 2, and one tenth of the one-time cost is added to it. At 0% and 100% the mask of DiSHE is empty or full, so its selection time is set to zero, while the CGD probe, which still sets the weights, is kept. Client phases run sequentially in one process: a phase time is the mean over clients, reconstruction is timed at one representative client, and the protocol time is the sum of the phases listed in Section 6.3. Initialization, key creation, evaluation, and network transfer are not timed. All runs used one workstation with an NVIDIA GeForce RTX 5090 GPU, an Intel Core i7-14700K CPU, 64 GB of DDR5 memory, and a 1 TB Samsung 980 Pro SSD; software versions are pinned in the released code. All methods share our framework and serializers, so the comparison describes these implementations, not lower bounds.

Table 5: Payload per round (MB) and protocol time per round (s) at a 10% ratio, with the payload split by component. Time includes consensus (MaskCrypt 0.51 s, ParaAegis 13.40 s) and one tenth of the one-time mask generation (FedML-HE 2.09 s, FAS 0.02 s).

Method	Plaintext	Ciphertext	Metadata	Mask msgs	Setup/10	Total	Time
DiSHE	391.0	573.6	0.4	–	0.1	965.2	1.51
MaskCrypt	402.3	459.8	894.0	44.7	–	1,800.7	1.82
FedML-HE	402.3	459.8	894.0	–	459.6	2,215.6	3.15
FAS	402.3	459.7	894.0	–	–	1,756.0	3.44
ParaAegis	402.3	459.8	894.0	268.2	–	2,024.2	15.54

Bytes. For client i the counted upload is

$$B_i^{\mathrm{up}} = \sum_{c \in \mathcal{E}_i} \left|\mathrm{ser}(\mathsf{ct}_{i,c})\right| + 4b\,(C - K) + M_i^{\mathrm{up}},$$

where M_i^{up} is the serialized length of the flags and float32 scores; the raw scores alone occupy $4C = 10{,}984$ bytes. The implementation sends plaintext chunks as float32. For one receiver the downlink is $B^{\mathrm{down}} = \sum_{c \in \mathcal{U}^t} |\mathrm{ser}(\overline{\mathsf{ct}}_c)| + 4b\,(C - |\mathcal{U}^t|) + M^{\mathrm{down}}$, where M^{down} serializes the flags. The total of a round is $\sum_i B_i^{\mathrm{up}} + N\,B^{\mathrm{down}}$ plus one tenth of the setup payload. The logged ciphertext bytes correspond to about 206 KB per fresh ciphertext and 131 KB per aggregated, rescaled one, sizes that reproduce the ciphertext bytes at every ratio to within 0.01%. Delivery of the initial model, key distribution, transport framing, and retransmissions are excluded. We use $1\,\mathrm{MB} = 10^6$ bytes. With everything encrypted, the 2,746 one-chunk ciphertexts of DiSHE take slightly less protocol time than the 2,729 densely packed ciphertexts of the baselines (9.09 s against 9.30–11.41 s) and 0.6% more ciphertext bytes; a model with $b < S$ would instead leave ciphertexts partly empty (Appendix E).

G RUNS AND DATA

Table 6: Runs reported in this paper, each for DiSHE and the four baselines. All use five fully participating clients, $E = 5$, batch size 128, Adam with learning rate 10^{-3}, Dirichlet concentration 0.5, and, for DiSHE, a probe of 512 examples processed in micro-blocks of 256.

Group	Model	Data	α	Rounds	Seeds	n_i
Accuracy	ResNet-18-GN	CIFAR-10	0.10	10	123, 456, 789	10,000
Accuracy	ResNet-18-GN	Fashion-MNIST	0.10	10	123, 456, 789	2,000
Accuracy	ResNet-18-GN	SVHN	0.10	10	123, 456, 789	2,000
Cost	ResNet-18-GN	CIFAR-10	0, 0.05, 0.10, 0.25, 0.75, 1	3	123	2,000
Attacks	LeNet-5	Fashion-MNIST	0.05	10	123, 456, 789	2,000

Table 6 lists the runs. **Preprocessing and training.** Inputs are float32 in $[0, 1]$ with a deterministic initial shuffle; there is no augmentation or per-channel normalization. Local training traverses each client's partition in its stored order, without per-epoch reshuffling or a learning-rate schedule; an epoch is 16 Adam steps for 2,000 examples and 79 for 10,000, including the last partial batch. Adam uses $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, no weight decay, and is reset whenever a client loads the incoming model. Global test accuracy is evaluated after every round with batch size 1,024; we report the value after the last round.

Partitioning. The allocator first assigns each class across clients by a Dirichlet draw with concentration 0.5 and then adjusts each client to the requested size by truncation or by borrowing examples from other clients. Borrowing can reuse an example already assigned elsewhere, so the partitions have exact sizes but are not guaranteed to be disjoint or to cover every training example. Shared data-loading code and common seeds do not by themselves certify that the baselines saw identical client sample identities or initializations; we compare methods at the level of configuration, not of paired runs.

H BASELINE IMPLEMENTATIONS

All four baselines run in the same framework as DiSHE, with the same models, local Adam training (five epochs, batch size 128, learning rate 10^{-3}, reset each round), clients, rounds, and seeds in every run group (Table 6). They are reimplementations; the original papers differ in threat model, keys, and training setup, and we describe the deviations below.

MaskCrypt (Hu & Li, 2025). Each client scores coordinates by the elementwise product of its local loss gradient and the difference between the exposed and the trained weights, and the ranked proposals are merged by interleaving and de-duplication into a common mask every round. The original protocol uses per-client keys and returns decrypted aggregate components to the server; our implementation uses the shared-key framework of this paper. Aggregation uses data-size weights on local models.

FedML-HE (Jin et al., 2023). Before training, each client computes a sensitivity map from averaged absolute per-example cross-entropy gradients; the maps are aggregated and the top fraction forms a mask that stays fixed for all rounds. The original paper defines sensitivity through a mixed derivative of parameter gradients with respect to outputs and discusses single-key and threshold variants. Aggregation uses data-size weights on local models.

FAS (local adaptation) (Korkmaz & Rao, 2025). A seeded, uniformly random mask covering the requested fraction of the coordinates is fixed across clients and rounds. The complementary update $v_i = P\Delta_i$ is clipped as a whole to L2 norm $C_{\text{clip}} = 1$, perturbed with Gaussian noise, and bitwise-scrambled; the server reverses the scrambling before aggregating, and clients reverse the downlink transformation. The published FAS adds Laplace noise; our adaptation uses Gaussian noise with the calibration below and protects model updates.

ParaAegis (local implementation) (Wu et al., 2025). Each client proposes the coordinates with the largest absolute update, and a vote forms a common mask each round; the complement is clipped and perturbed as for FAS. The ratio decay is 1.0, and the logs of the accuracy runs record a 10% encrypted fraction in every round. The original protocol also protects the proposed coordinate indices

during the vote; we measure the vote exchange as implemented (268 MB per round at a 10% ratio) and do not claim the same protection.

Noise setting (not a DP guarantee). Both perturbed baselines use

$$\sigma = \frac{\min_i(2C_{\text{clip}}/n_i)}{\epsilon}\sqrt{2qT\log(1/\delta)}, \qquad C_{\text{clip}} = 1,\ \epsilon = 1,\ \delta = 10^{-5},\ q = 1,\ T = 10,$$

which gives $\sigma = 0.003034854$ for 10,000 examples per client and 0.015174271 for 2,000, the values recorded in all accuracy runs. This formula uses a sensitivity of $2C_{\text{clip}}/n_i$, which would hold for an average of n_i per-example contributions each clipped to C_{clip} (Abadi et al., 2016). Our adaptations instead clip the whole multi-epoch Adam update, whose ℓ_2 sensitivity to one example (or one client) can be as large as $2C_{\text{clip}}$; under the same formula σ would have to be n_i times larger, and the data-dependent ParaAegis mask and the ten rounds would also have to be accounted for. The stated (ϵ, δ) is therefore nominal: we claim no differential-privacy guarantee for these adaptations, and their results are not at a privacy level matched to anything. With five equal clients, the root-mean-square norm of the aggregate noise over the unencrypted coordinates, $\sigma\sqrt{0.9d/5}$, is about 4.30 (CIFAR-10) and 21.52 (Fashion-MNIST, SVHN), against at most 1 for the clipped unencrypted aggregate. With clipping coefficient $c_i = \min\{1, C_{\text{clip}}/\|v_i\|\}$ and noise $\xi_i \sim \mathcal{N}(0, \sigma^2 I)$, the transmitted update is $H\Delta_i + c_i v_i + P\xi_i$ with $H = I - P$ the HE mask, and the data-size-weighted aggregate differs from that of the unperturbed updates by $\sum_i a_i(c_i - 1)v_i + \sum_i a_i P\xi_i$. Under independent client noise each unencrypted aggregate coordinate has noise variance $\sigma^2 \sum_i a_i^2$ ($\sigma^2/5$ for five equal clients), and the expected squared noise norm over d_P unencrypted coordinates is $d_P\sigma^2 \sum_i a_i^2$.

Table 7: Clipping diagnostics of the perturbed baselines, from the logs of all completed accuracy runs (150 client-rounds per row: 3 seeds × 10 rounds × 5 clients). Every complementary update was clipped; the encrypted part is not clipped.

Method	Data	Mean clipping coefficient	Min–max	Noise SD σ
FAS	CIFAR-10	0.02441	0.02041–0.02954	0.00303485
FAS	Fashion-MNIST	0.04581	0.03704–0.05911	0.01517427
FAS	SVHN	0.04298	0.03481–0.05930	0.01517427
ParaAegis	CIFAR-10	0.03193	0.02713–0.03690	0.00303485
ParaAegis	Fashion-MNIST	0.05870	0.04446–0.07253	0.01517427
ParaAegis	SVHN	0.05323	0.04072–0.06883	0.01517427

Comparability. DiSHE budgets chunks (with the mandatory head and rounding of Equation (2)), while the baselines budget coordinates, so a nominal ratio of 0.10 does not match encrypted coordinates, ciphertexts, bytes, or protection across methods. Applying a common shared-key framework changes the original information flow of MaskCrypt and FedML-HE; differences between methods should not be attributed to selection alone.

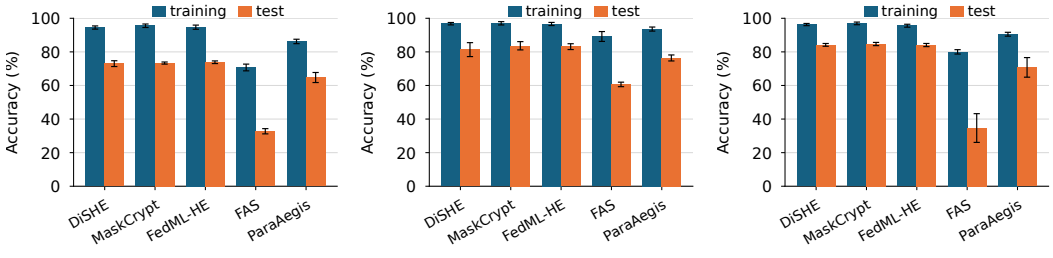
I Accuracy statistics

Let $a^{\text{train}}_{s,i,t}$ be client i's accuracy on its own training partition after local training in round t of run s ($S = 3$ seeds, $N = 5$ clients, $T = 10$ rounds), and g_s the test accuracy of the decrypted global model after the last round of run s. The two reported metrics are

$$\mu_{\text{global}} = \frac{1}{3}\sum_s g_s, \qquad s_{\text{global}} = \Big(\frac{1}{2}\sum_s (g_s - \mu_{\text{global}})^2\Big)^{1/2},$$

$$\mu_{\text{local}} = \frac{1}{5}\sum_i z_i, \qquad s_{\text{local}} = \Big(\frac{1}{4}\sum_i (z_i - \mu_{\text{local}})^2\Big)^{1/2},$$

with $z_i = \frac{1}{3}\sum_s \max_t a^{\text{train}}_{s,i,t}$. The local mean equals the average of all 15 client–run maxima, but s_{local} is the standard deviation across the five client-index means, not across runs, and client indices in different seeded partitions need not correspond to the same data. Neither deviation is a confidence interval or a standard error. Table 8 lists the differences in final global test accuracy.



(a) CIFAR-10 (b) Fashion-MNIST (c) SVHN

Figure 3: Accuracy with ResNet-18-GN, five non-IID clients, ten rounds, and a nominal 10% encryption ratio (data of Table 1). *test*: final global test accuracy, mean and sample deviation over three seeds. *training*: mean best-round local training accuracy, with the deviation across client-index means. The two bars are not a generalization gap.

Table 8: Final global test accuracy of DiSHE minus that of each comparator (percentage points; positive means DiSHE has the higher mean). Descriptive differences of three-seed means, not significance tests.

Comparator	CIFAR-10	Fashion-MNIST	SVHN
MaskCrypt	-0.35	-2.30	-0.51
FedML-HE	-0.83	-1.77	$+0.10$
FAS	$+40.26$	$+20.72$	$+49.51$
ParaAegis	$+8.26$	$+4.96$	$+13.40$

J ATTACK PROTOCOLS AND FURTHER RESULTS

Evaluation rationale. Gradient-inversion results depend on the attacker's observations and assumptions (Huang et al., 2021; Balunović et al., 2022). We deliberately expose target gradients without accumulation over multiple local training steps. For a single SGD step, $\Delta = -\eta g$, so a known learning rate makes the update equivalent to a scaled gradient. This motivates our direct gradient-access evaluation: DLG, IG, and iDLG use individual examples, while iLRG uses batches. We retain the encryption masks and attacker model estimates from the saved checkpoints. The experiment therefore tests protection under these masks without relying on the difficulty introduced by multi-epoch local training.

Checkpoints and observation. With ten rounds and a capture interval of four, checkpoints are written in rounds 0, 4, and 8, after local training and mask selection and before aggregation. A checkpoint separates the true round-entry model θ^t, used only to generate targets, from the attacker's pre-update estimate of the model, which for DiSHE is the visible-update estimate $\hat{\theta}^t$ of Equation (7); in round 0 the two coincide. For each seed, three training examples are drawn once (attack seed 2026) from the pooled client data for DLG and IG, together with three batches of up to 128 examples for iLRG, and the same targets are used for every method. For DiSHE they are reused under each of the five client masks and each captured round, so a target need not belong to the client whose mask is applied. The attacker observes the target gradient only on the coordinates that the attacked client sends in the clear: for DiSHE the plaintext chunks of that client, for the baselines the complement of their common mask, clipped and perturbed with the noise of our implementations for FAS and ParaAegis. The objectives index only visible coordinates and never pull hidden ones toward zero. A cell of Table 2 thus averages 45 reconstructions for DiSHE (five clients, three targets, three seeds) and 9 for each baseline; these are not independent samples.

Reduction and metrics. Table 9 lists the attack settings. Among restarts we keep the one with the lowest attacker objective, never the one closest to the ground truth. Image metrics use the $[0, 1]$ scale, and PSNR uses unit peak value. We report SSIM, LPIPS, MSE (Table 10), and PSNR of every reconstruction and no success threshold. Besides the mean over clients, Table 11 reports the most exposed client of DiSHE: for each seed, round, attack, and label setting, the client with the lowest mean LPIPS over the three targets (ties broken by the higher SSIM), whose values are then averaged

Table 9: Reconstruction settings, shared by all methods. Both attacks start from a dummy model equal to the attacker's estimate, clamp pixels to $[0, 1]$ after each step, and stop early after 200 steps without an improvement of 10^{-7}. With the label unknown, a soft label is optimized jointly with the image.

	DLG	IG
Objective	$\sum_{\text{visible}} (g'(x,y) - g^\star)^2 + \tau\,\mathrm{TV}_{\mathrm{sum}}(x)$	$1 - \frac{\langle g'_V, g^\star_V \rangle}{\|g'_V\| \, \|g^\star_V\| + 10^{-12}} + \tau\,\mathrm{TV}_{\mathrm{mean}}(x)$
Optimizer	Adam, learning rate 0.01	Adam on the sign of the input gradient, learning rate 0.1
Iterations, restarts	up to 1,000; 3	up to 4,800; 1
TV weight τ	10^{-4}	10^{-4}

over seeds. Selected reconstructions from the most resistant client are shown separately in Figure 4. Reconstructions with the label known and unknown come from the same checkpoints.

Table 10: MSE and PSNR of the reconstructions of Table 2 (higher MSE and lower PSNR mean a worse reconstruction).

	DLG						IG					
	MSE			PSNR (dB)			MSE			PSNR (dB)		
Method	0	4	8	0	4	8	0	4	8	0	4	8
DiSHE	.027	.133	.138	16.7	9.6	9.0	.048	.097	.098	14.0	11.0	11.0
DiSHE, label unknown	.032	.130	.142	16.2	9.2	9.1	.045	.086	.095	14.2	11.7	11.0
FAS	.094	.160	.162	10.4	8.3	8.1	.168	.128	.127	7.8	9.2	9.2
FedML-HE	.034	.169	.176	15.8	7.9	7.7	.086	.162	.135	11.5	8.2	9.0
MaskCrypt	.030	.196	.187	16.1	7.6	7.5	.047	.183	.178	13.7	7.7	7.6
ParaAegis	.098	.170	.152	10.2	8.0	8.4	.170	.202	.203	7.8	7.1	7.0

Table 11: SSIM and LPIPS for the most exposed client of DiSHE (lowest mean LPIPS per seed, round, attack, and label setting), in the layout of Table 2.

	DLG						IG					
	SSIM ↓			LPIPS ↑			SSIM ↓			LPIPS ↑		
Method	0	4	8	0	4	8	0	4	8	0	4	8
DiSHE	.391	.136	.091	.177	.462	.496	.252	.264	.255	.326	.374	.377
DiSHE, label unknown	.426	.080	.085	.169	.515	.518	.247	.240	.187	.349	.384	.418

Label attacks. iDLG reads the label from the signs of the visible classifier gradients and needs a unique candidate. iLRG estimates class embeddings from visible classifier weight and bias gradients ($h_k \approx G_W[k,:]/G_b[k]$), computes probabilities $\mathrm{softmax}(0.01\,(Wh_k + b))$ with the attacker's classifier, and solves the visible bias-gradient equations $\sum_k n_k p_k[r] - n_r = B\,G_b[r]$ together with $\sum_k n_k = B$ by a pseudoinverse, followed by clipping and rounding to the batch size B. Both attacks need the final-layer gradients, which every client of DiSHE encrypts; iLRG also needs the final-layer bias, which the FedML-HE mask hides. Table 12 lists the outcomes, where a blocked attempt counts as no recovery. With the label unknown, the jointly optimized labels of DLG are correct in 100.0, 66.7, and 55.6% of the DiSHE cases at rounds 0, 4, and 8, and those of IG in 100.0, 100.0, and 93.3%.

Table 12: Analytic label attacks at rounds 0, 4, and 8 (three targets or batches per round and seed; five clients for DiSHE). Recall and F1 refer to the recovered set of batch labels and are undefined when iLRG cannot run.

Method	iDLG correct	iLRG runs	iLRG instance recall	iLRG label F1
DiSHE	0 / 135	0 / 135	–	–
FAS	9 / 27	27 / 27	0.711	0.899
FedML-HE	27 / 27	0 / 27	–	–
MaskCrypt	27 / 27	27 / 27	0.701	0.894
ParaAegis	9 / 27	27 / 27	0.698	0.896

J.1 QUALITATIVE RECONSTRUCTION EXAMPLES

Figure 4 shows two examples selected after inspecting the saved reconstructions. For each seed, round, attack, and label setting, we select the most resistant DiSHE client by its highest mean LPIPS over the three targets. Ties are broken by lower mean SSIM and then lower client index. This selects client 3 in both label settings for each row. These examples complement the client averages in Table 2 and the most exposed client results in Table 11.

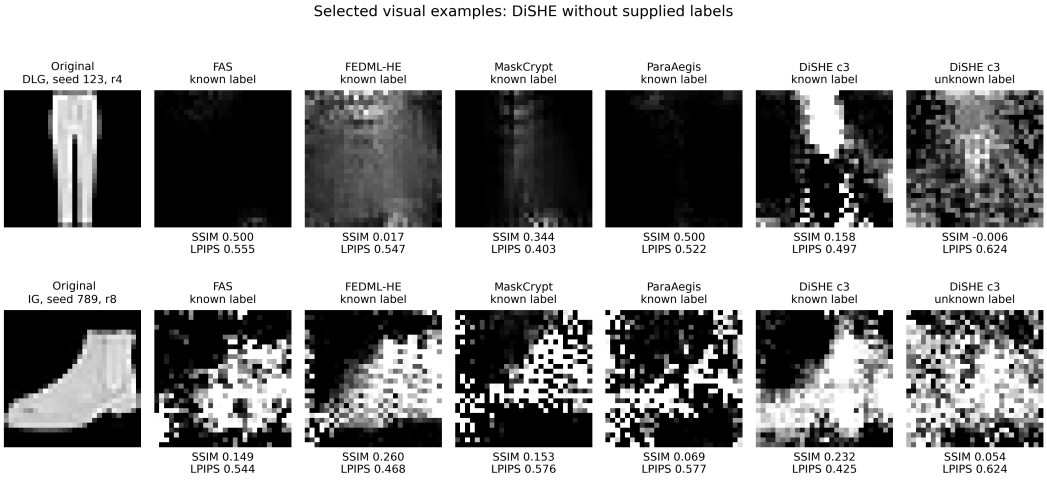


Figure 4: Selected gradient reconstructions on Fashion-MNIST with LeNet-5 at a nominal 5% encryption ratio. The top row shows DLG at round 4 with seed 123; the bottom row shows IG at round 8 with seed 789. Within each row, all methods attack the same target, shown on the left. The four baseline columns use the true label. The final two columns show DiSHE with the true label supplied and with a soft label optimized jointly with the image. The classifier head remains encrypted in both DiSHE settings. Each panel is a single saved reconstruction, displayed on the same $[0, 1]$ grayscale range. Lower SSIM and higher LPIPS indicate poorer reconstruction.

In both examples, DiSHE with an unknown label has lower SSIM and higher LPIPS than the displayed baseline reconstructions. The ankle-boot outline visible in the known-label IG result is largely lost when the label is optimized jointly with the image. These cases show that withholding the label can substantially weaken reconstruction. The baseline columns use supplied labels, and the DiSHE clients were selected for strong resistance. The aggregate results in Tables 2 and 11 provide the broader comparison across clients and seeds.