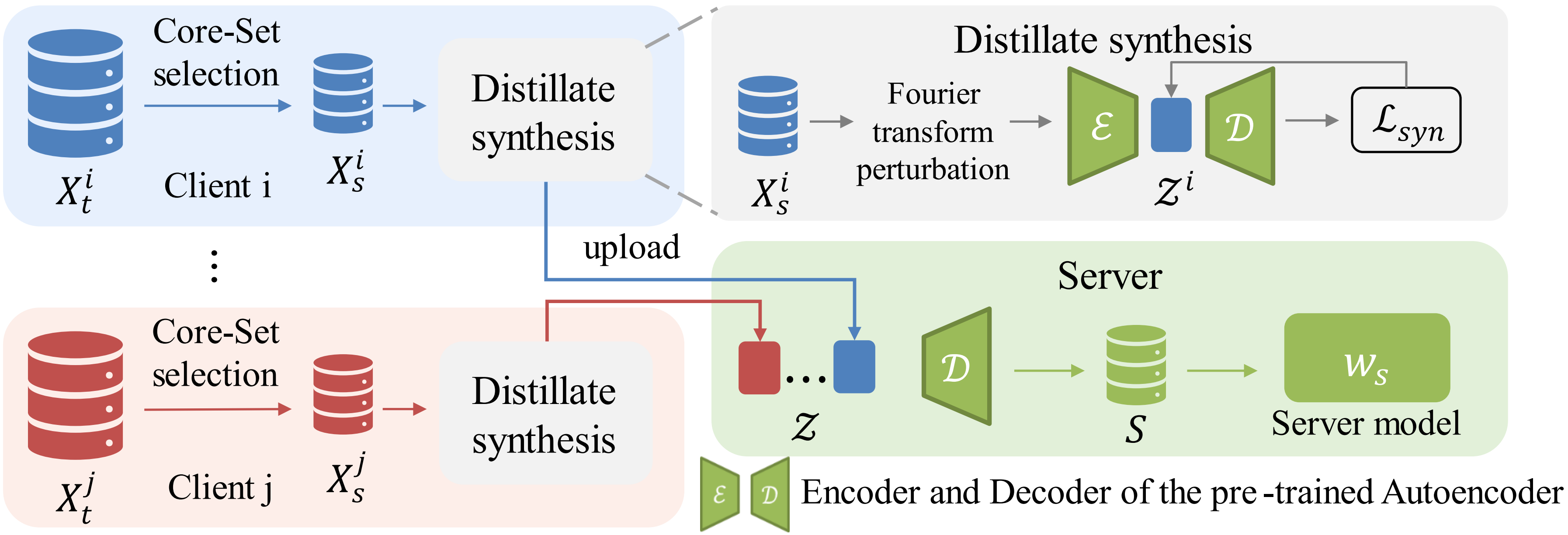
One-shot Federated Learning via Synthetic Distiller-Distillate Communication

Junyuan Zhang, Songhua Liu, Xinchao Wang
National University of Singapore



Background

Challenges and current solution for one-shot FL

- **Data heterogeneity**: Data varies among institutions (amount, quality, imaging equipment/parameters, etc.), resulting in inconsistent client models.
- Methods based on weight aggregation fall short in accuracy.
- Methods based on Data-free knowledge distillation demonstrate comparable results.

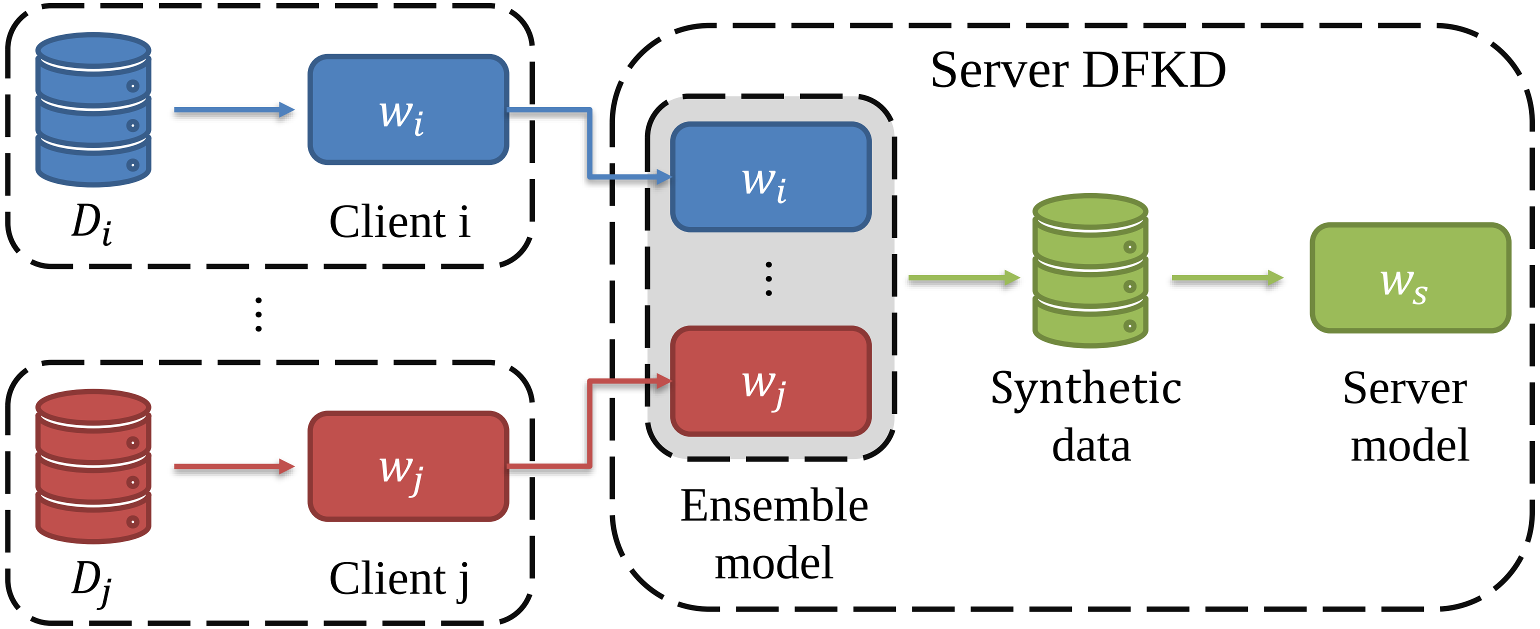


Fig. DFKD one-shot FL

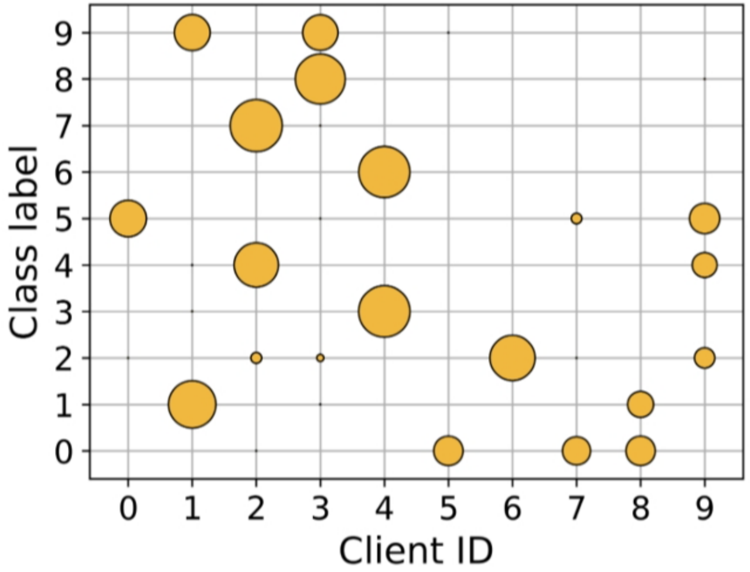
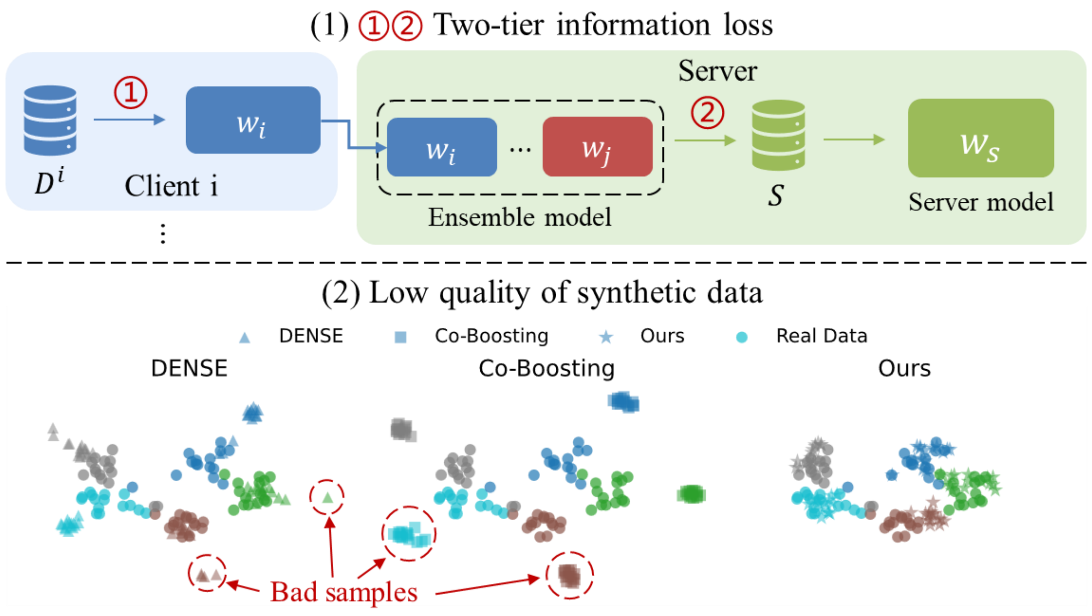


Fig. Data heterogeneity

Challenges for DFKD one-shot FL

- **Two-tier information loss**:
  - ① Local training (from local data to client model)
  - ② Data synthesis (from ensemble model to inversed data)
- **Low quality of synthetic data**:
  - Inconsistent client models caused by data heterogeneity, obscure correct predictions.

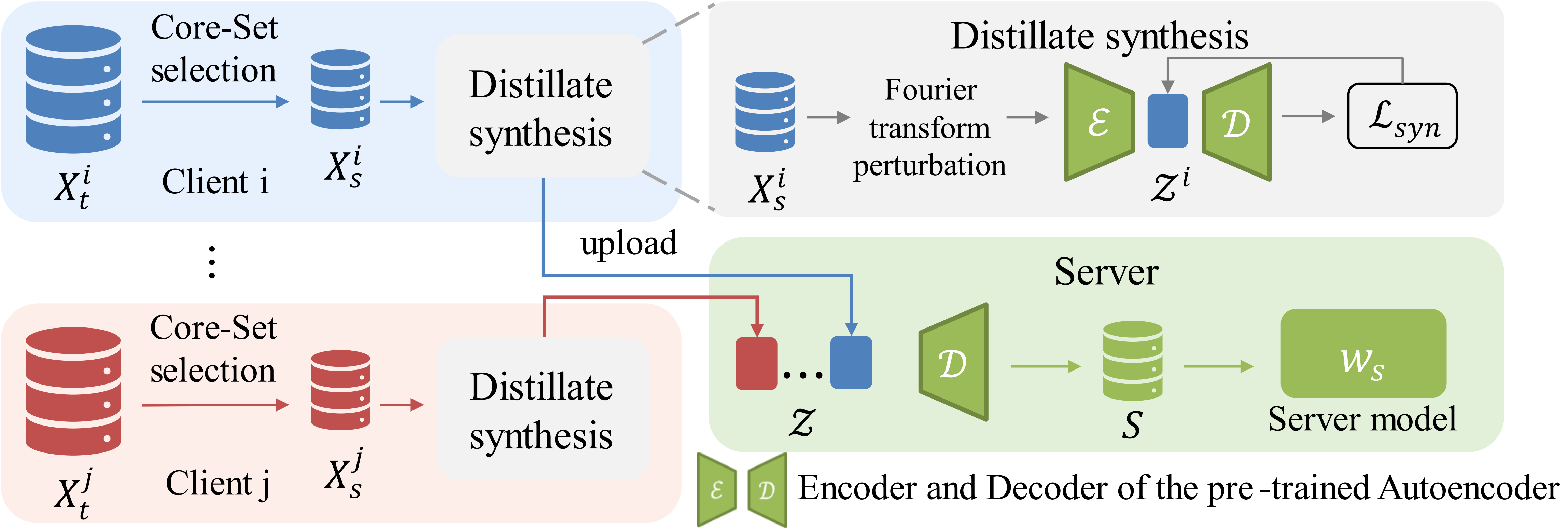


Method

We propose to directly transmit synthetic data

- **First Stage: Core-Set selection** to extract diverse and informative Core-Set from clients' local data domains.

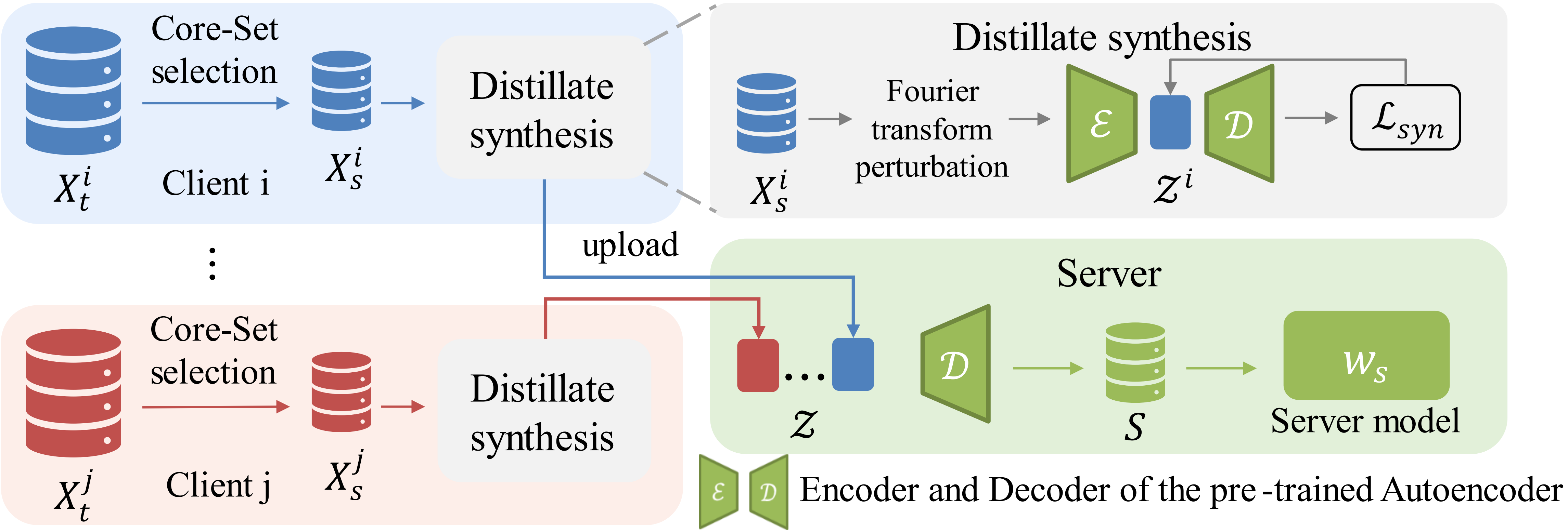
- **Second Stage**: **Distillate synthesis** to synthesize informative, privacy-enhanced, and communication-efficient distillates for server-side training.



Method

We propose to directly transmit synthetic data

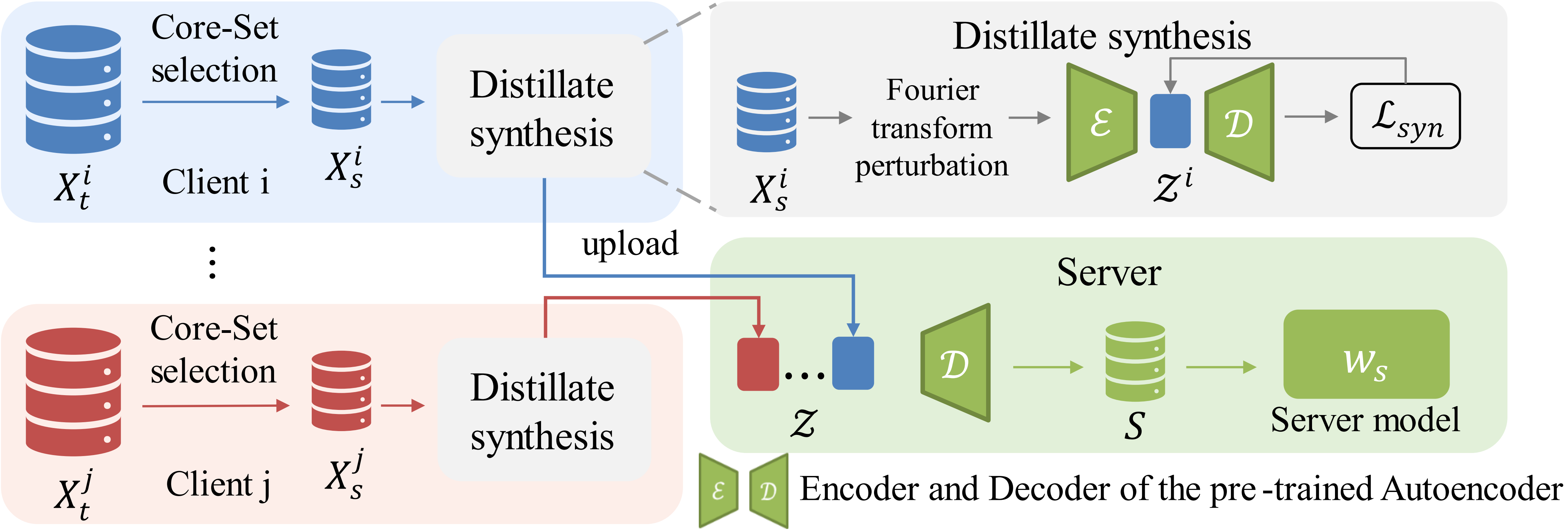
- **First Stage: Core-Set selection** to extract diverse and informative Core-Set from clients' local data domains.
  - Find the Core-Set with highest information entropy: $(X_s, Y_s) = \arg\max_{X,Y} I_{\mathcal{V}}(X_t \rightarrow Y_t)$
  - Use local model h as observer $\mathcal{V}$ and compute score s : $s = -\mathcal{L}(h(x), y)$



Method

We employ two techniques to further distill the Core-Set into distillates

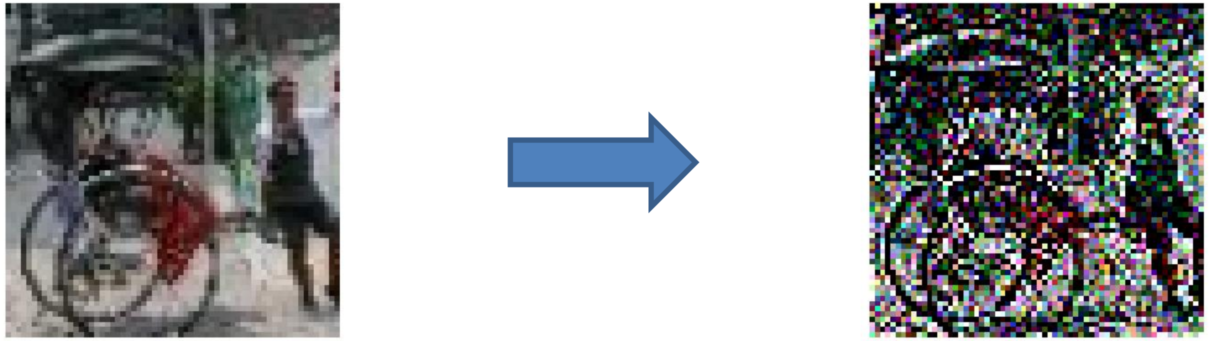
- **Second Stage**: **Distillate synthesis** to synthesize informative, privacy-enhanced, and communication-efficient distillates for server-side training.
- **1): Distillate initialization with Fourier transform perturbation**
- **2): Distillate synthesis with pre-trained Autoencoders**

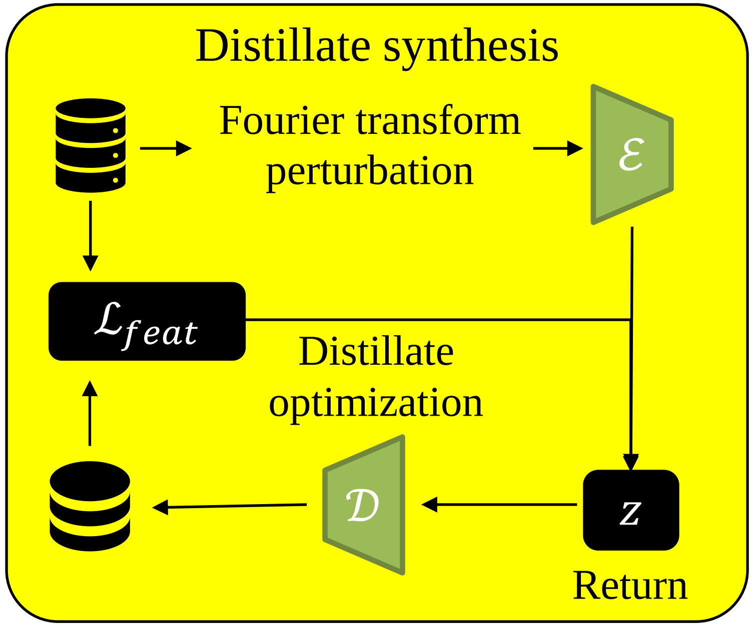


Method

We employ two techniques to further distill the Core-Set into distillates

- **1): Distillate initialization with Fourier transform perturbation**: We alter the amplitude component of Core-Set sample, **reducing privacy information** while preserving semantic content.
  - Fourier transform on Core-Set sample: $\mathcal{F} = \mathcal{A}(x) \times e^{-j \times \mathcal{P}(x)}$
  - Perturb the amplitude information via linearly interpolating: $\hat{\mathcal{A}}(x) = (1-\lambda)\mathcal{A}(x) + \lambda\mathcal{A}(x^*)$
  - Combine the perturbed amplitude spectrums with the original phase component and use inverse Fourier transform $\mathcal{F}^{-1}(\cdot)$ to generate the perturbed Core-Set sample: $x = \mathcal{F}^{-1}(\hat{\mathcal{A}}(x) \times e^{-j \times \mathcal{P}(x)})$





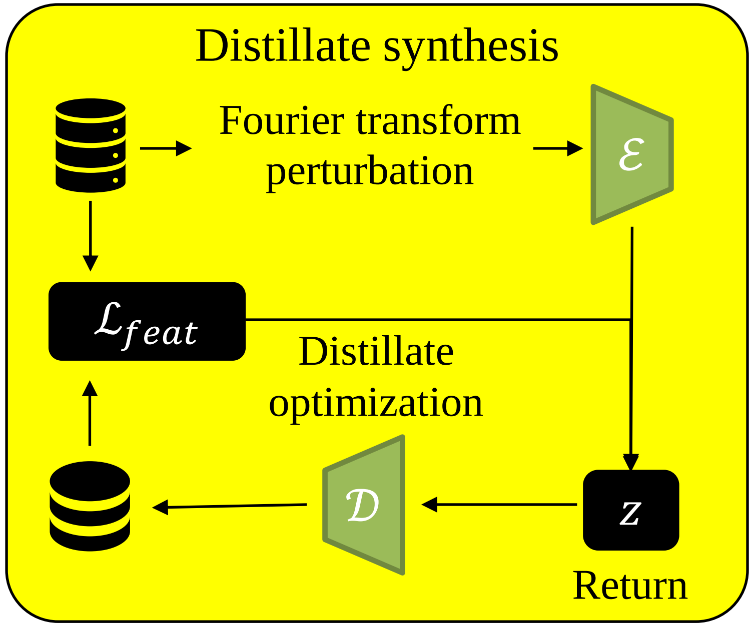
Method

We employ two techniques to further distill the Core-Set into distillates

- **2): Distillate synthesis with pre-trained Autoencoders**: We employ a pre-trained Autoencoder to distill the perturbed Core-Set into **generalizable distillates, simultaneously reducing communication costs**.
  - Encoding perturbed sample with a pre-trained Autoencoder: $z = \mathcal{E}(x)$
  - Learn a latent z which is as close as possible to the corresponding data in the Core-Set:

$$\arg\min_{z} \left\| h(\mathcal{D}(z)) - h(x) \right\|^2$$

  - Send the synthetic data to sever for training.



Experiments: General Results

General experimental results under various Data heterogeneity settings

Model	Methods	ImageNette $\alpha=0.1$	ImageNette $\alpha=0.3$	ImageNette $\alpha=0.5$	Tiny-ImageNet $\alpha=0.1$	Tiny-ImageNet $\alpha=0.3$	Tiny-ImageNet $\alpha=0.5$	OpenImage -
ConvNet	Central		89.60			49.73		33.61
	FedAVG	10.68±0.23	10.04±0.10	9.83±0.27	-	-	-	3.08±0.17
	F-DAFL	44.95±0.72	52.23±0.23	58.34±0.55	5.25±0.41	8.89±0.61	10.28±0.10	3.36±0.56
	DENSE	42.09±0.68	48.64±1.91	54.74±0.75	11.45±0.08	14.69±0.48	15.15±0.22	7.00±0.84
	Co-Boosting	39.36±0.70	56.15±1.33	**58.60±1.02**	6.66±0.35	9.81±0.26	10.75±0.11	13.59±0.98
	FedSD2C	**50.68±0.20**	**57.89±0.96**	58.17±0.51	**20.73±0.12**	**23.53±0.18**	**24.10±0.30**	**23.00±0.24**
ResNet-18	Central		90.00			61.98		34.17
	FedAVG	9.86±0.13	10.06±0.20	10.76±0.35	-	-	-	1.68±0.16
	F-DAFL	37.86±0.38	39.52±0.46	46.06±0.16	7.91±0.22	12.30±0.36	13.31±0.56	12.75±0.14
	DENSE	38.37±0.36	47.85±2.17	49.78±2.11	8.88±0.23	13.05±0.36	17.24±0.43	14.85±0.62
	Co-Boosting	27.06±0.61	28.53±0.86	30.53±1.12	10.29±0.43	14.35±0.93	16.39±0.59	9.52±1.52
	FedSD2C	**47.52±0.51**	**53.69±0.17**	**55.90±0.53**	**26.83±0.10**	**29.92±0.37**	**31.66±0.85**	**22.69±0.14**

Tab. Accuracy of different one-shot FL methods over three datasets with ConvNet and ResNet-18. indicates. We vary the $\alpha = \{0.1, 0.3, 0.5\}$ to simulate different levels of data heterogeneity

- FedSD2C surpasses all other methods in most settings and demonstrates the independence from model structures

Experiments: Privacy Evaluation

Privacy-preserving techniques	ImageNette				Tiny-ImageNet			
	ConvNet↑	ResNet-18↑	PSNR↓	SSIM↓	ConvNet↑	ResNet-18↑	PSNR↓	SSIM↓
-	51.87	51.82	-	-	22.62	28.29	-	-
Ours($\lambda = 0.1$)	51.26	50.55	23.48	73.20	22.03	28.22	20.54	54.89
Ours($\lambda = 0.5$)	51.36	48.97	19.97	64.23	21.77	28.09	18.06	44.18
Ours($\lambda = 0.8$)	50.68	47.52	16.42	50.80	20.85	26.83	16.95	35.89
$Laplace(s = 0.2, p = 0.1)$	48.61	45.25	24.02	81.66	21.50	27.48	22.25	73.09
$Gaussian(s = 0.2, p = 0.1)$	48.31	46.70	24.82	85.89	21.48	27.51	23.38	78.90
$Laplace(s = 0.2, p = 0.2)$	45.61	38.01	20.05	73.13	19.32	23.66	19.99	64.51
$Gaussian(s = 0.2, p = 0.2)$	45.81	38.09	20.30	76.11	19.32	23.52	20.35	68.56
FedMix	41.86	37.76	16.88	58.93	13.86	16.26	16.43	56.91

Model Inversion Attack

Table S3: Membership Inference Attack.

Method	TPR@FPR=0.1%
Sharing model-based methods (DENSE, Co-Boosting)	22.81
FedSD2C	20.13

Membership Inference Attack

- FedSD2C achieve comparable privacy protection with minimal performance degradation

Experiments: Effectiveness of VAE

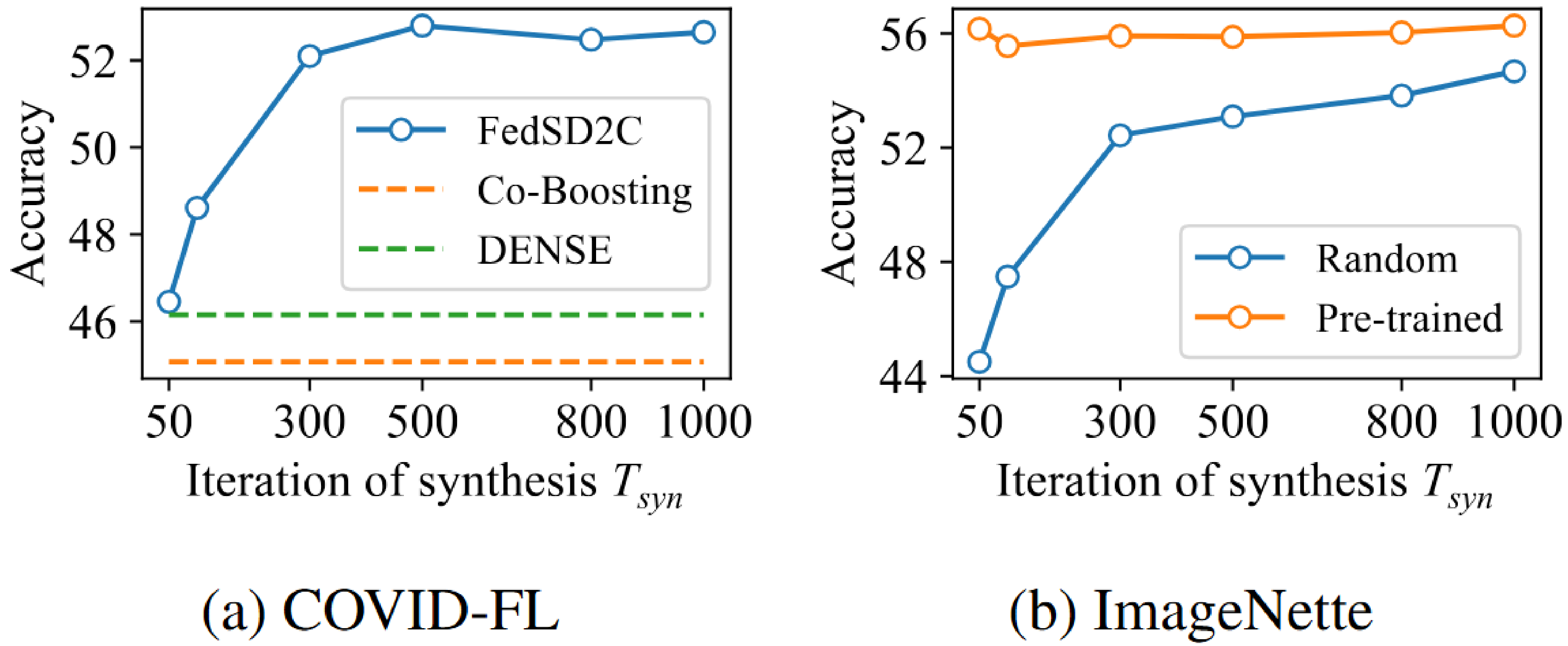


Figure 3: (a) Experiments on the medical image data domain.Adopting pre-trained Autoencoders on other data domains canreduce performance. However, this can be mitigated by increasingTsyn. (b) Experiments of FedSD2C with randomly initializeddownsampling and upsampling modules (blue line) compared topre-trained Autoencoders (orange line) on ImageNette. Withoutpre-trained knowledge, FedSD2C requires a higher Tsyn for distil-late synthesis but can still achieve comparable results. ResNet-18is used for both experiments.

Thank you!