PCEvolve: Private Contrastive Evolution for Synthetic Dataset Generation via Few-Shot Private Data and Generative APIs

Jianqing Zhang[1,2] Yang Liu[3] Jie Fu[4] Yang Hua[5]

Tianyuan Zou[2] Jian Cao[1] Qiang Yang[3]

1 

2 

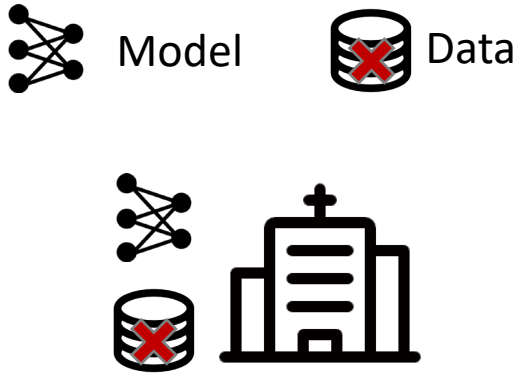
3 

4 

5 

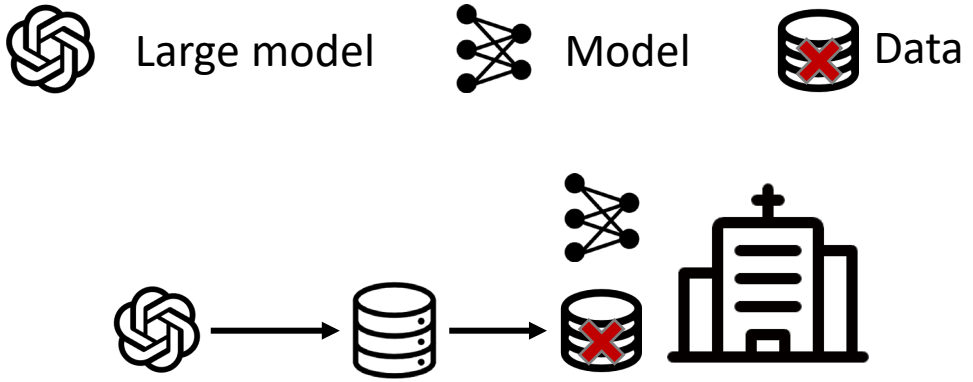
Background: Data scarcity

- Data scarcity challenges AI model training, especially for specialized domains like medicine (e.g., *pneumonia recognition*) and industry (e.g., *anomaly detection*).
  - Low data quality
  - Limited data access
  - …



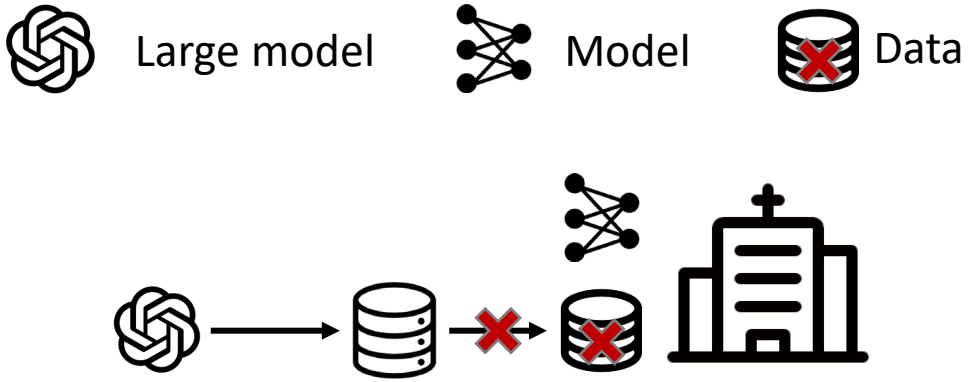
Background: Large models can help ?

- With pre-trained knowledge, large models can generate *synthetic data* to alleviate data scarcity.
  - Using various generative APIs
  - Prompt engineering
  - …



Background: Large models cannot help

- However, large models suffer from specialized domains[1].
  - Domain gap between the synthetic and private data
  - Personalized requirements
  - …

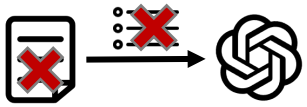


[1] Johri, Shreya, et al. "An evaluation framework for clinical use of large language models in patient interaction tasks." *Nature Medicine* (2025): 1-10.

Background: Large models cannot help

- Widely-used approaches in specialized domains for large models:
  - Fine-tuning
    - **Costly** for large model training, **data scarcity**
  - Few-shot in-context learning (ICL)
    - **Privacy issue**, effortful **prompt engineering**
  - Zero-shot ICL + selection
    - **Costly** for large amount data generating, effortful **prompt engineering**

Fine-tuning

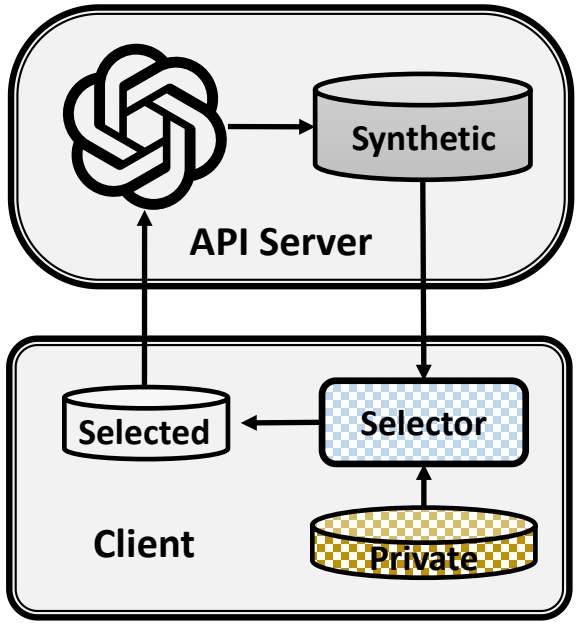


Few-shot ICL

Zero-shot ICL

Private Evolution (PE)

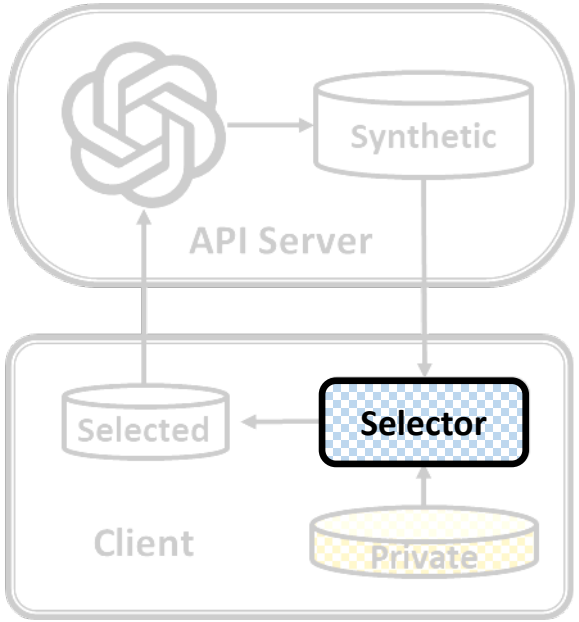
- PE = few-shot ICL + selection + evolution
  - Using *private data* for scoring the quality of synthetic data to enable selection & evolution
  - Like AlphaEvolve[1], FunSearch[2], etc.



[1] Novikov, Alexander, et al. “AlphaEvolve: A Gemini-Powered Coding Agent for Designing Advanced Algorithms.” Google DeepMind, 14 May 2025.
[2] Romera-Paredes, Bernardino, et al. "Mathematical discoveries from program search with large language models." Nature 625.7995 (2024): 468-475.

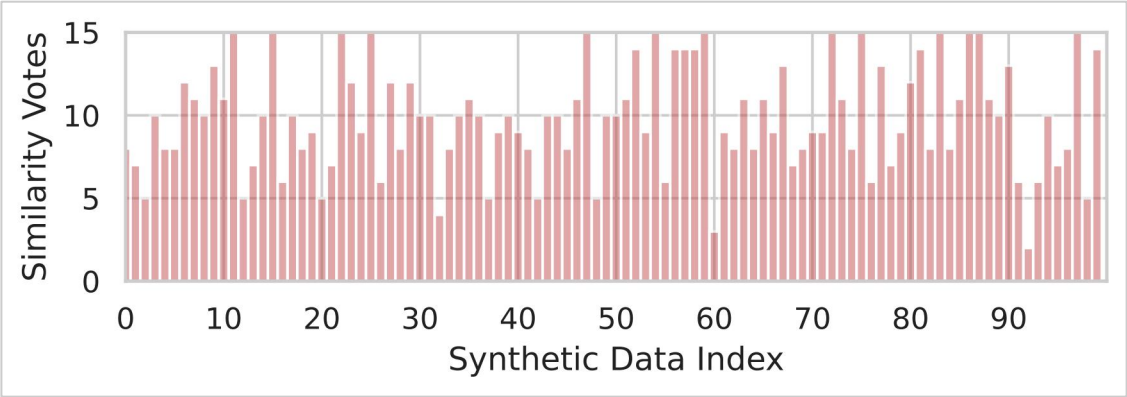
Private Evolution (PE)

- PE = few-shot ICL + selection + evolution
  - The key is the **selector**, it is the engine



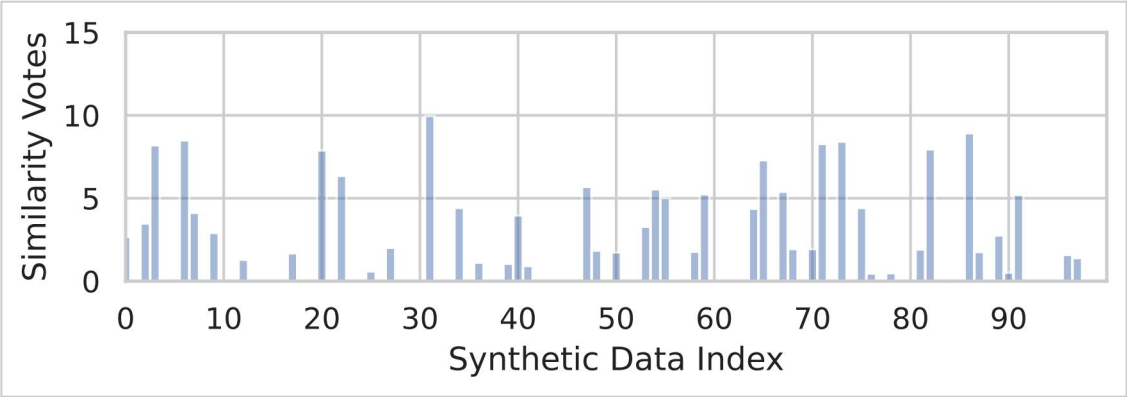
Private Evolution's shortcomings

- Selector's voting mechanism: let private data to vote on the quality of synthetic data
  - More votes, more informative
  - Suitable with massive private data
  - E.g., 1000 private data points



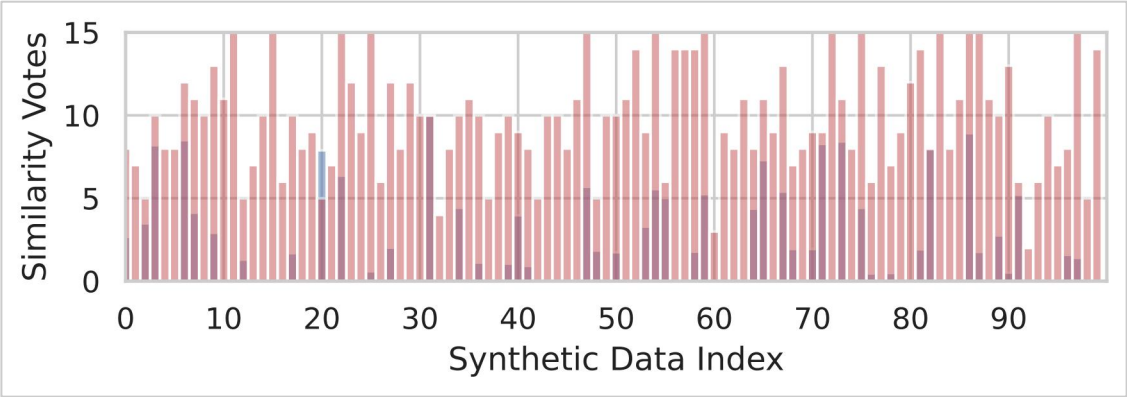
Private Evolution's shortcomings

- Selector's Differential Privacy (DP): add proper **Gaussian noise** to the votes



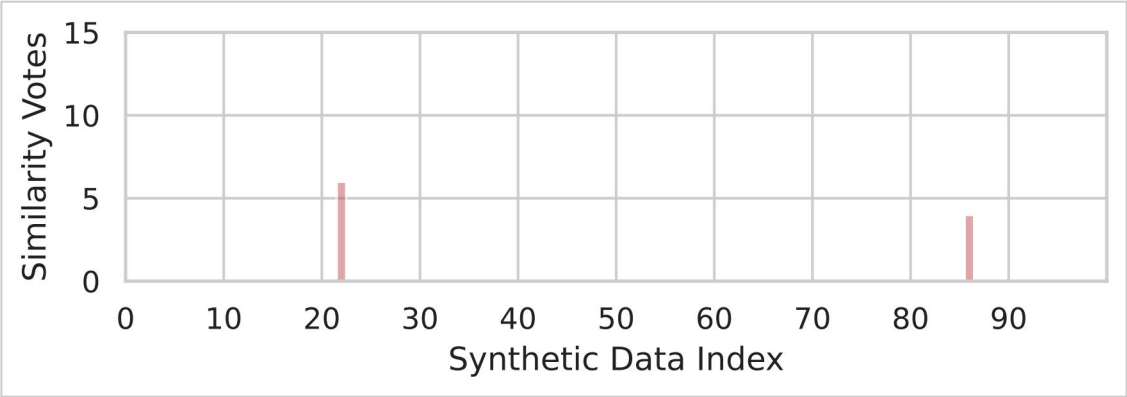
Private Evolution's shortcomings

- Selector's DP: add proper **Gaussian noise** to the votes
  - More votes, less influenced
  - E.g., 1000 private data points



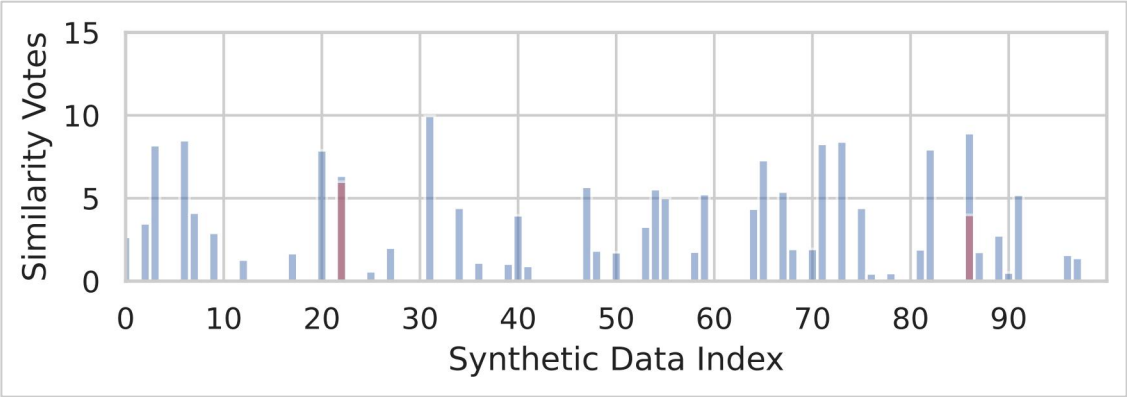
Private Evolution's shortcomings

- Selector's voting mechanism can only produce a few votes given few-shot private data
  - Less votes, less informative
  - E.g., 10 private data points



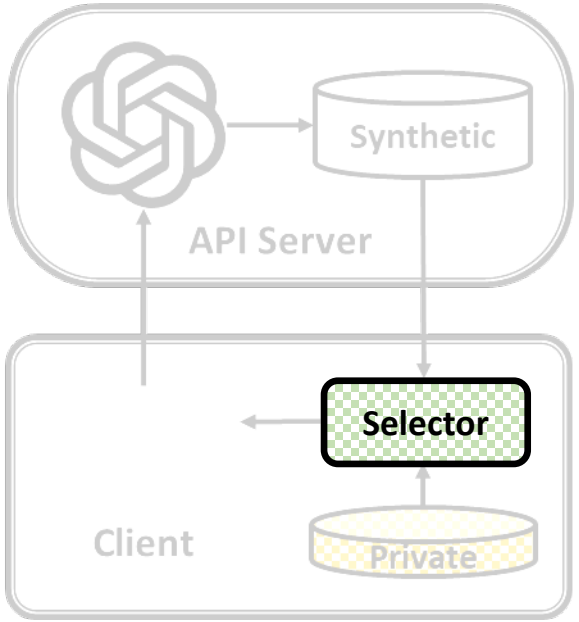
Private Evolution's shortcomings

- Selector's voting mechanism can only produce a few votes given few-shot private data
  - Less votes, more influenced
  - Gaussian mechanism is sensitive to the private data amount
  - E.g., 10 private data points



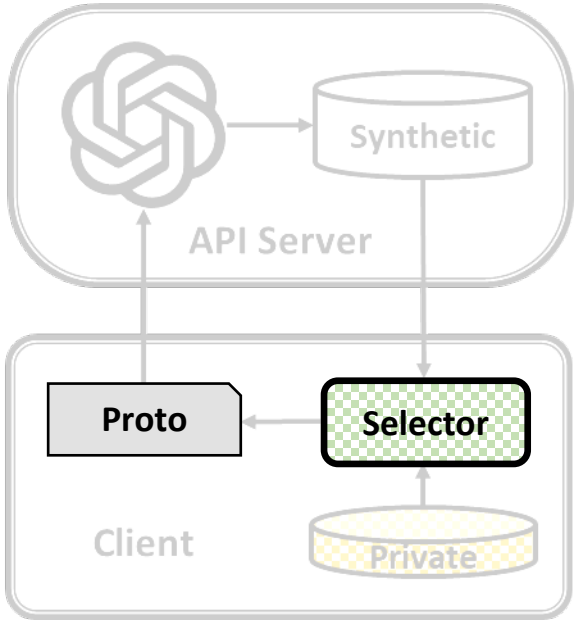
Our Private Contrastive Evolution (PCEvolve)

- Abandon PE's voting mechanism and **Gaussian noise**
- Adapt the **Exponential Mechanism (EM)** to our scenario for DP
  - Agnostic to private data amount



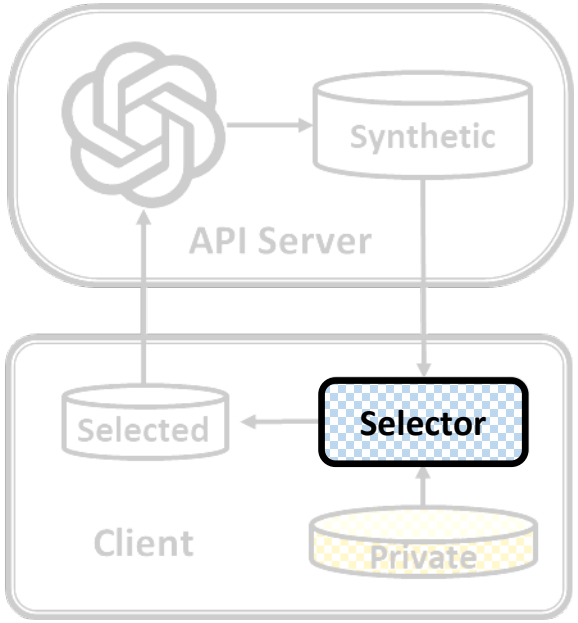
Our Private Contrastive Evolution (PCEvolve)

- Devise a **contrastive filter** to exploit inter-class relationships inside few-shot private data
- Select **prototypical synthetic data** for feedback

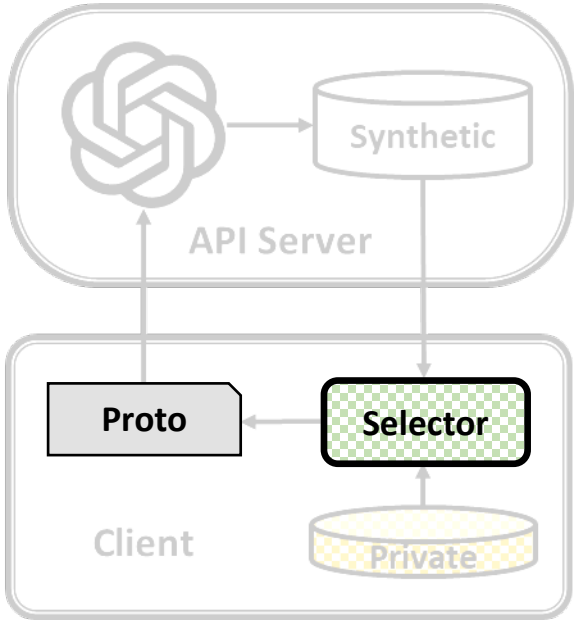


Our Private Contrastive Evolution (PCEvolve)

- The difference between PE and our PCEvolve in the framework



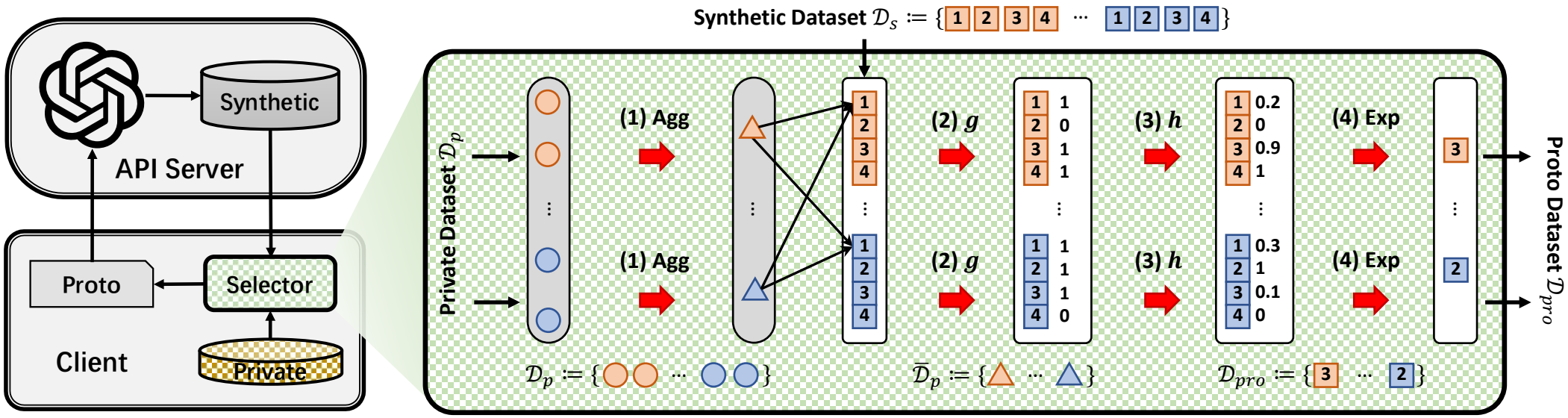
PE



PCEvolve

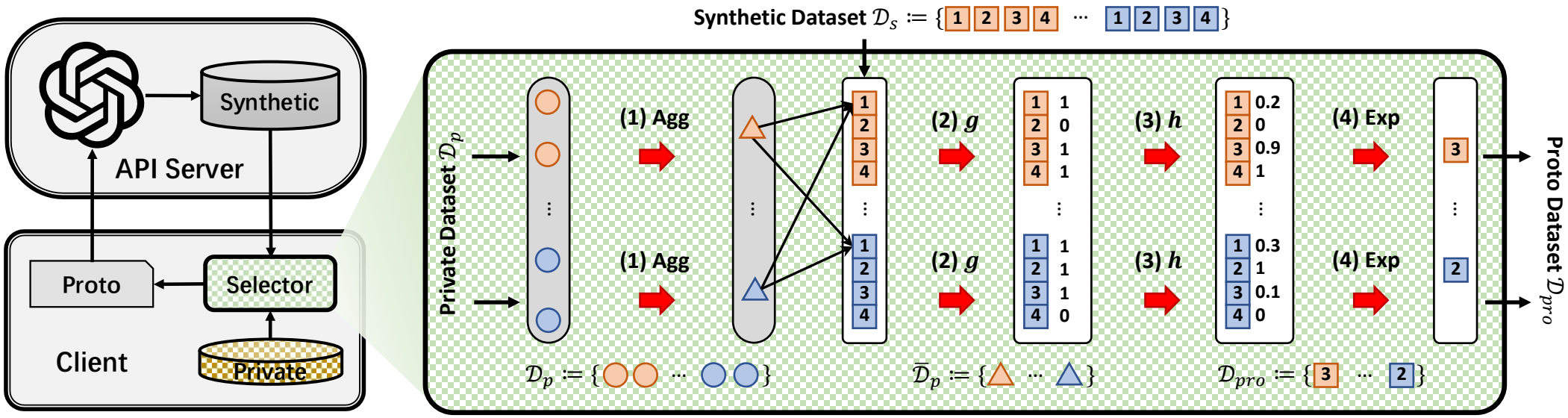
Overview of PCEvolve

- Different colors denote distinct data classes
- “Agg”: class center aggregation | “Exp”: Exponential Mechanism (EM) $\mathcal{M}_u\ (u = h \circ g)$



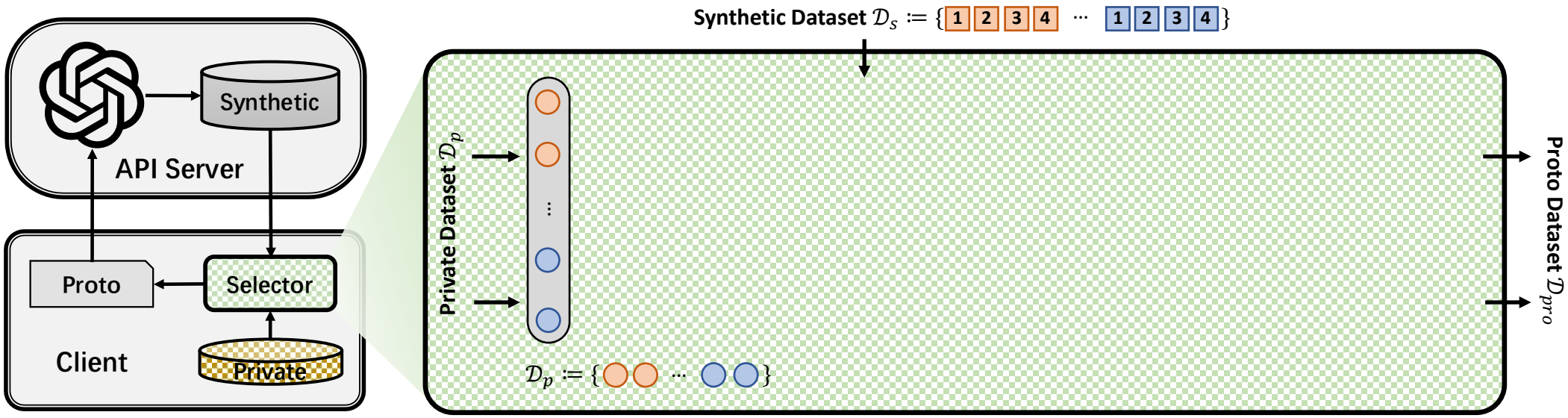
Overview of PCEvolve

- g: contrastive filter (introduced later)
- h: similarity calibrator (introduced later)



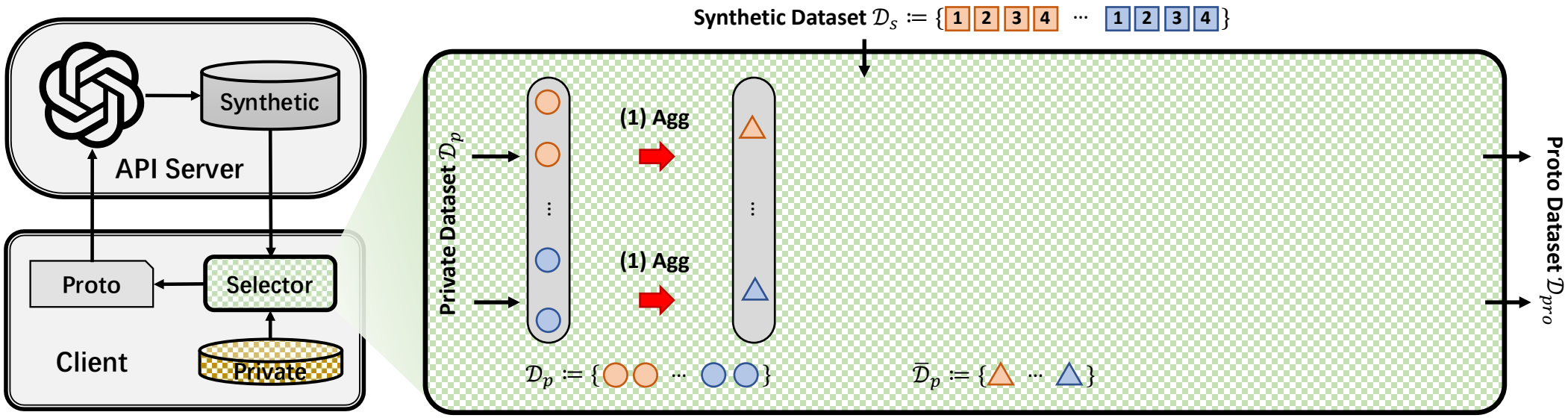
Step-by-step

- In each evolving iteration, we have the synthetic dataset (with indices) and private dataset
- Selector should output the prototypical dataset



Step-by-step (1) Agg

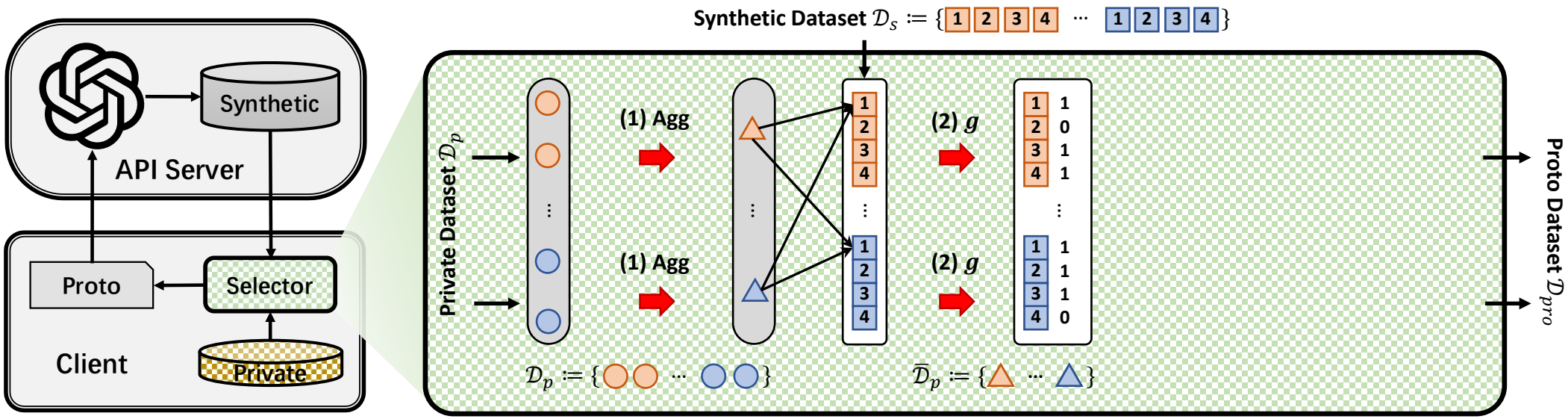
- Aggregate few-shot private data to obtain the private center set $\bar{\mathcal{D}}_p := \{\bar{d}_p^c\}_{c\in[C]}$



Step-by-step (2) contrastive filter g

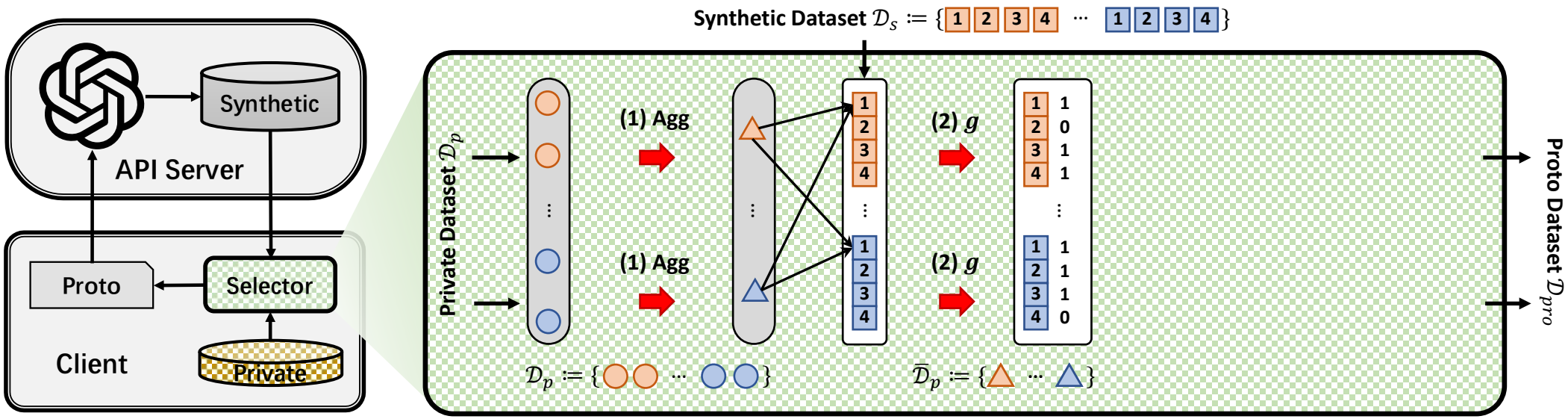
- Select synthetic data that can be correctly classified into their corresponding classes using $\bar{\mathcal{D}}_p$ as class identifiers to exploit the inter-class information

$$g(d_s^c, \bar{\mathcal{D}}_p) := \begin{cases} 1, & \text{if } \ell_2(d_s^c, \bar{d}_p^c) < \min_{c'} \{\ell_2(d_s^c, \bar{d}_p^{c'})\} \\ 0, & \text{otherwise} \end{cases}$$



Step-by-step (3) similarity calibrator h

- Due to the large domain gap, optimizing only discriminability using g won't narrow the gap
- We need to narrow down the similarity between synthetic and private data

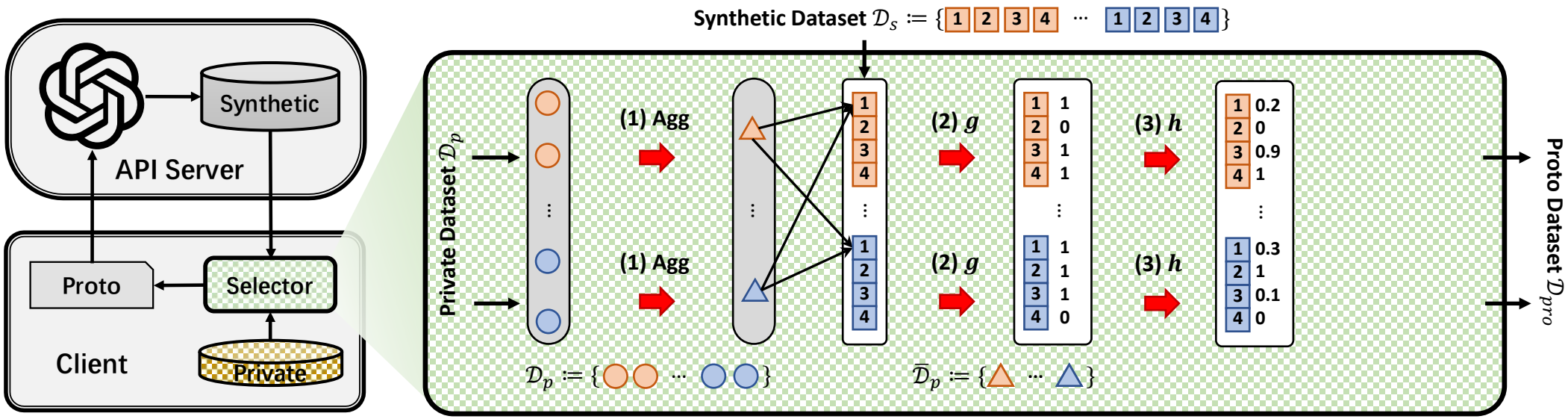


Step-by-step (3) similarity calibrator h

- Thus, we devise h as

$$h(d_s^c, \bar{\mathcal{D}}_p) := \begin{cases} e^{-\ell_2(d_s^c, \bar{\mathcal{D}}_p)}, & \text{if } g(d_s^c, \bar{\mathcal{D}}_p) = 1 \\ 0, & \text{otherwise} \end{cases}$$

- The range of ℓ_2 is $[0, +\infty)$, so $h \in [0,1]$, making $u = h \circ g \in [0,1]$ and the sensitivity $\Delta_u = 1$



Step-by-step (3) similarity calibrator h

- In fact, ℓ_2 rarely spans the full range of $[0, +\infty)$, but still we have $u = h \circ g \in [0,1]$ and $\Delta_u = 1$
  - A waste of range ---> a **waste** of privacy budget for DP
- To fully use the range of h at each time, we calibrate the similarity by

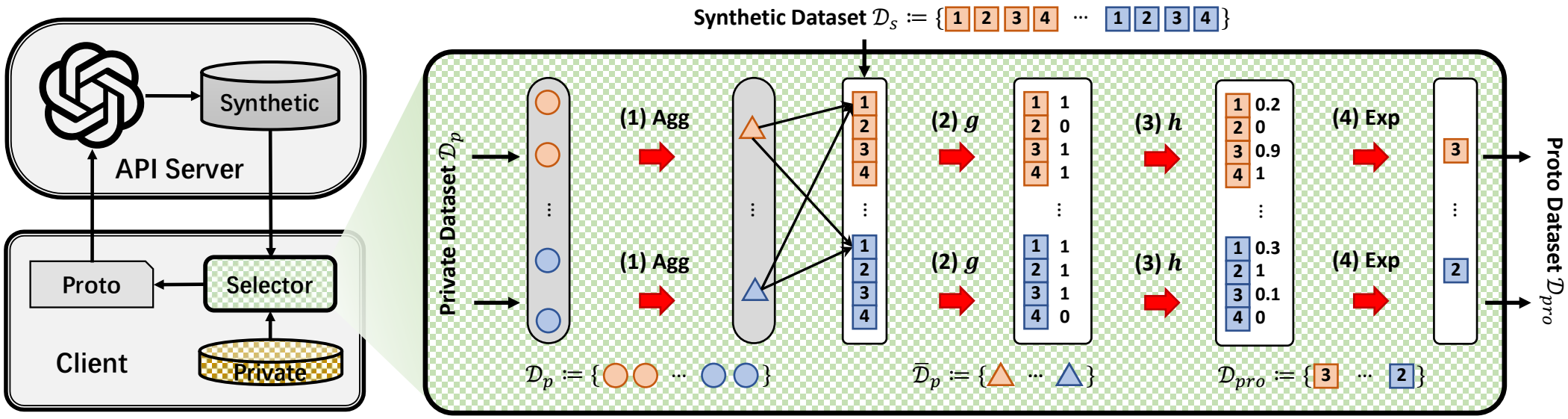
$$h(d_s^c, \bar{\mathcal{D}}_p) := \begin{cases} e^{-\frac{\ell_2(d_s^c, \bar{\mathcal{D}}_p) - \ell_{\min}^c}{\ell_{\max}^c - \ell_{\min}^c} \cdot \tau}, & \text{if } g(d_s^c, \bar{\mathcal{D}}_p) = 1 \\ 0, & \text{otherwise} \end{cases}$$

$$\text{s.t.} \quad \ell_{\max}^c := \max_{d_s^c \in \mathcal{D}_s^c,\ g(d_s^c, \mathcal{D}_p)=1} \ell_2(d_s^c, \bar{\mathcal{D}}_p),$$

$$\ell_{\min}^c := \min_{d_s^c \in \mathcal{D}_s^c,\ g(d_s^c, \mathcal{D}_p)=1} \ell_2(d_s^c, \bar{\mathcal{D}}_p),$$

Step-by-step (4) Applying $\mathcal{M}_u$

- Given a well-defined u, we can apply EM $\mathcal{M}_u$, and
- The best candidate with the highest u value has the greatest probability of being selected



Step-by-step (4) Applying $\mathcal{M}_u$

- Given a well-defined u, we can apply EM $\mathcal{M}_u$, and
- The best candidate with the highest h value has the greatest probability of being selected

Definition 3.3 (Exponential Mechanism (Dong et al., 2020; McSherry & Talwar, 2007)). Given a parameter ϵ, an arbitrary range $\mathcal{R}$, and a utility function $u : \mathbb{X} \times \mathcal{R} \to \mathbb{R}$ with sensitivity $\Delta_u := \max_{r \in \mathcal{R}} \max_{\mathcal{D},\mathcal{D}'} |u(\mathcal{D}, r) - u(\mathcal{D}', r)|$, a randomized algorithm $\mathcal{M}_u$ is called the Exponential Mechanism (EM), if the outcome r is sampled with probability proportional to $\exp\left(\frac{\epsilon \cdot u(\mathcal{D}, r)}{2\Delta_u}\right)$ and $\mathcal{M}_u$ is ϵ-DP:

$$Pr[\mathcal{M}_u(\mathcal{D}) = r] = \frac{\exp\left(\frac{\epsilon \cdot u(\mathcal{D}, r)}{2\Delta_u}\right)}{\sum_{r' \in \mathcal{R}} \exp\left(\frac{\epsilon \cdot u(\mathcal{D}, r')}{2\Delta_u}\right)}.$$

Step-by-step (4) Applying $\mathcal{M}_u$

- Given a well-defined u, we can apply EM $\mathcal{M}_u$, and
- The best candidate with the highest h value has the greatest probability of being selected

Definition 3.4 (Sequential Composition (Dwork et al., 2006)). Given any mechanism $\mathcal{M}_1(\cdot)$ that satisfies ϵ_1-DP, and $\mathcal{M}_2(s, \cdot)$ that satisfies ϵ_2-DP for any s, then $\mathcal{M}(\mathcal{D}) = \mathcal{M}_2\left(\mathcal{M}_1(\mathcal{D}), \mathcal{D}\right)$ satisfies $(\epsilon_1 + \epsilon_2)$-DP .

Remark 3.5. Given the same ϵ, $\mathcal{M}_u$ (ϵ-DP) provides stronger privacy protection than $\mathcal{M}_\sigma$ ((ϵ, δ)-DP) for $\delta > 0$.

PCEvolve is ϵ_*-DP

- **Theorem 4.1.** Algorithm 1 PCEvolve satisfied ϵ_*-DP.

Algorithm 1 `PCEvolve`

Input: Private dataset $\mathcal{D}_p$, i2i API G_{i2i}, t2i API G_{t2i}, text prompt $\mathcal{T}$, total privacy cost ϵ_*, number of class C, number of iteration T, encoder E_f, and similarity calibrating factor τ.

Output: Synthetic dataset $\mathcal{D}_s$.

1: $\mathcal{D}_s^0 \leftarrow G_{t2i}(\mathcal{T})$ and $\epsilon = \frac{\epsilon_*}{T \cdot C}$.
2: **for** evolution iteration $t = 1, \ldots, T$ **do**
3: **for** class $c = 1, \ldots, C$ **do**
4: Get u scores for $\mathcal{D}_s^{t,c}$ via Eq. (1) and Eq. (3).
5: Get $\mathcal{D}_{pro}^{t,c}$ by sampling data from $\mathcal{D}_s^{t,c}$, with the index set $\mathcal{R}$ of $\mathcal{D}_s^{t,c}$ and probabilities

$$Pr[\mathcal{M}_u(\mathcal{D}_p) = r \in \mathcal{R}] = \frac{\exp\left(\frac{\epsilon \cdot u(\mathcal{D}_p, r)}{2\Delta_u}\right)}{\sum_{r' \in \mathcal{R}} \exp\left(\frac{\epsilon \cdot u(\mathcal{D}_p, r')}{2\Delta_u}\right)}.$$

6: $\mathcal{D}_s^t \leftarrow G_{i2i}(\mathcal{T}, \mathcal{D}_{pro}^t)$, where $\mathcal{D}_{pro}^t = \{\mathcal{D}_{pro}^{t,c}\}_{c=1}^{C}$.
7: **return** Synthetic dataset $\mathcal{D}_s^T$.

Comprehensive Experiments

- **Representative specialized domains**: medicine, industry
- **Various Image generation APIs**: online, offline
- **Various downstream models**: ResNets, Inception, ViT
- **Scaling law** of synthetic data
- …

Representative specialized domains

- **COVIDx**: chest X-ray images for COVID-19
- **Came17**: tumor tissue patches from breast cancer metastases
- **KVASIR-f**: endoscopic images for gastrointestinal abnormal findings detection
- **MVAD-l**: leather surface anomaly detection

Top-1 accuracy (%) on four specialized datasets

	COVIDx	Came17	KVASIR-f	MVAD-l
Init	49.34	50.47	33.43	33.33
RF	50.01	54.82	34.66	48.17
GCap	50.86	55.77	32.66	27.33
B	50.42	54.41	32.57	43.21
LE	50.02	55.44	35.51	27.93
DPImg	49.14	61.06	33.35	37.03
PE	59.63	63.66	48.88	57.41
PE-EM	57.60	63.34	43.01	50.06
PCEvolve-GM	56.91	62.63	43.55	55.56
PCEvolve	**64.04**	**69.10**	**50.95**	**59.26**

Various Image generation APIs

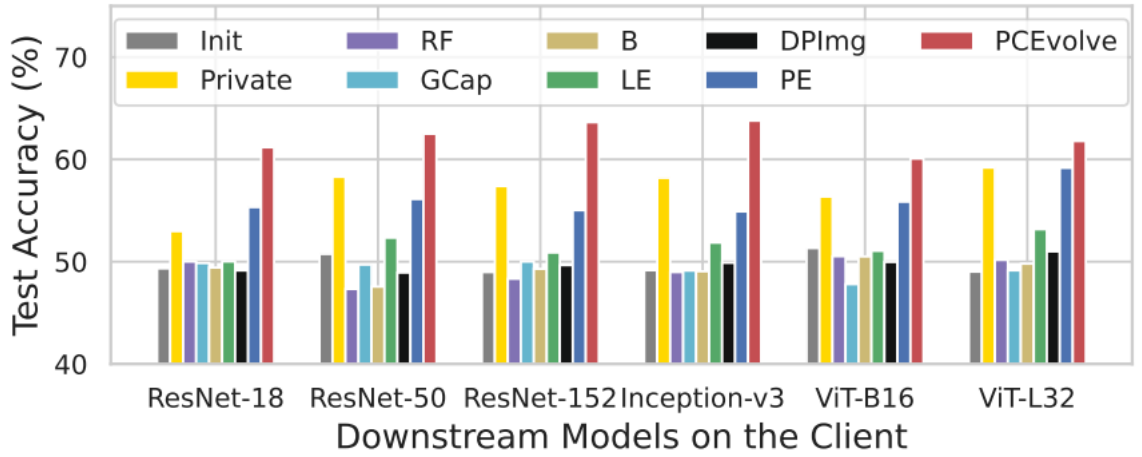
- **SD**: Stable Diffusion API
- **SD+IPA**: SD API with the IP-Adapter
- **OJ**: OpenJourney API

Top-1 accuracy (%) on COVIDx and KVASIR-f using SD+IPA and OJ (online) APIs

	COVIDx		KVASIR-f	
APIs	SD+IPA	OJ (online)	SD+IPA	OJ (online)
RF	45.03	47.91	27.22	36.55
GCap	53.70	47.42	28.77	37.11
B	46.61	50.22	26.27	36.61
LE	49.79	53.17	31.22	37.38
DPImg	50.58	49.61	36.89	35.05
PE	56.92	54.47	48.83	48.17
PCEvolve	**60.46**	**65.88**	**52.77**	**54.58**

Various downstream models

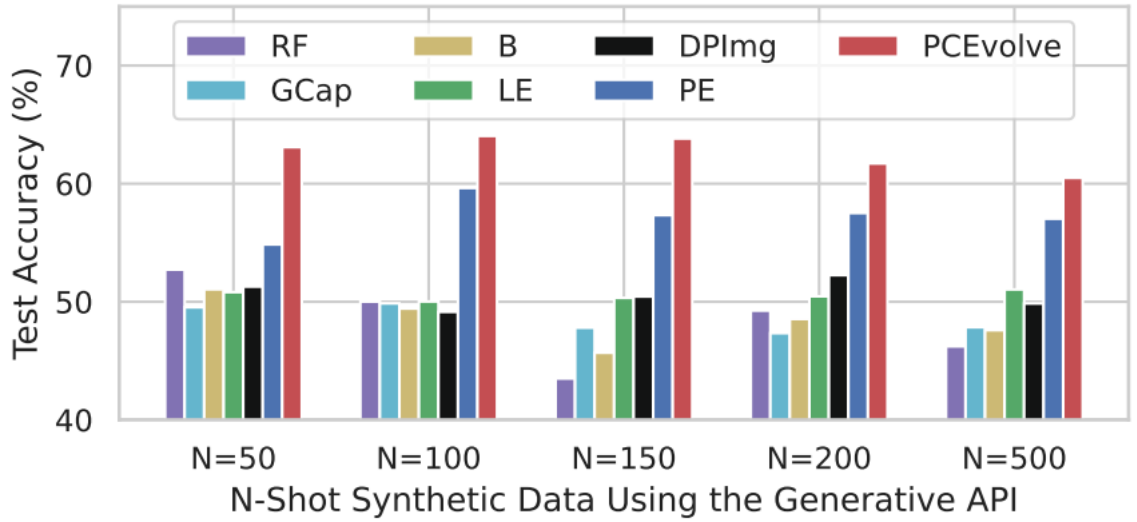
- Our PCEvolve consistently outperforms across all types of downstream models.



Top-1 accuracy of various downstream models on COVIDx. "Private" represents an additional private baseline, which directly trains downstream models on few-shot private data.

Scaling law of synthetic data

- The best N is between 100 and 150
- Increasing the amount of synthetic data introduces more noise from the API



Top-1 accuracy of ResNet-18 on COVIDx with varying synthetic data shots per class per iteration.

Synthetic images

(a) Initial (b) PE (c) PCEvolve (d) Private

Generated leather surface images w.r.t. MVAD-l for industry anomaly detection. The three rows show normal images, cut defects, and droplet defects. “Initial” denotes the initial synthetic images in PE and PCEvolve. “Private” denotes the real images from MVAD-l.

Feel free to contact me!

Home page: https://github.com/TsingZ0

Paper with code: https://github.com/TsingZ0/PCEvolve

Thanks!