This Appendix provides a detailed method and extensive experiments of the paper, KFC. Concretely,

$\triangleright$ Appendix A gives the problem definition and the proposed KFC method, including detailed descriptions of knowledge reconstruction and feedback consolidation.

$\triangleright$ Appendix B presents extensive experiments on datasets of MNIST, FashionMNIST, and CIFAR10.

A Detailed Method

Problem Definition. In the continual generative learning (CGL) setting, a sequence of T tasks is given, each task is characterized by a distinct dataset $\mathcal{D}^t = \{(\boldsymbol{x}_i^t, y_i^t) \mid i = 1, ..., N_t\}$, $t = 1, ..., T$, where $\boldsymbol{x}_i^t$ and $y_i^t \in \{1, ..., C^t\}$ indicate the i^{th} sample and its label in task t, respectively; N_t is the number of sample, and C^t is the number of class labels. Note that labels in tasks are disjoint, i.e., there is no overlapping class label between any two tasks (Belouadah et al., 2021).

Mathematically, CGL method aims to estimate the conditional probability distribution with parameters θ_t through observable variables y^t and unobserved variables $\boldsymbol{z}$, i.e., $p_\theta(\boldsymbol{x} \mid y) = \int \sum_{t=1}^{T} p_{\theta_t}(\boldsymbol{x}^t \mid y^t, \boldsymbol{z})\, p(\boldsymbol{z}) \mathrm{d}\boldsymbol{z}$ related to the whole tasks, where each task sequentially arrives. In other words, a given label y_i^t and a prior noise $\boldsymbol{z}$ could stimulate the generator θ_t to produce sample $\boldsymbol{x}_i^t$. The Evidence Upper BOund (EUBO) of $-\log p_{\theta_t}(\boldsymbol{x}^t \mid y^t)$ could be deduced with Jensen's inequality Ramapuram et al. (2020),

$$\begin{aligned}
-\log p_{\theta_t}(\boldsymbol{x}^t \mid y^t) &= -\log \int \frac{q_{\phi_t}(\boldsymbol{z} \mid y^t, \boldsymbol{x}^t)}{q_{\phi_t}(\boldsymbol{z} \mid y^t, \boldsymbol{x}^t)} p_{\theta_t}(\hat{\boldsymbol{x}}^t \mid y^t, \boldsymbol{z}) p(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \\
&\leq -\int q_{\phi_t}(\boldsymbol{z} \mid y^t, \boldsymbol{x}^t) \cdot \log \frac{p_{\theta_t}(\hat{\boldsymbol{x}}^t \mid y^t, \boldsymbol{z}) p(\boldsymbol{z})}{q_{\phi_t}(\boldsymbol{z} \mid y^t, \boldsymbol{x}^t)} \mathrm{d}\boldsymbol{z} \\
&= \underbrace{-\mathbb{E}_{q_{\phi_t}(\boldsymbol{z}|y^t, \boldsymbol{x}^t)}[\log p_{\theta_t}(\hat{\boldsymbol{x}}^t | y^t, \boldsymbol{z})] + \mathrm{KL}\left[q_{\phi_t}(\boldsymbol{z}|y^t, \boldsymbol{x}^t) || p(\boldsymbol{z})\right]}_{EUBO} \\
&= \underbrace{\mathcal{L}_{CVAE}^{R}(\phi_t, \theta_t) + \mathcal{L}_{\mathrm{CVAE}}^{V}(\phi_t)}_{EUBO}
\end{aligned} \tag{A1}$$

where $\mathbb{E}$ and KL are the expectation and Kullback–Leibler divergence operations, respectively. $\mathcal{L}_{\mathrm{CVAE}}^{R}(\phi_t, \theta_t)$ and $\mathcal{L}_{\mathrm{CVAE}}^{V}(\phi_t)$ are termed as the reconstruction loss and variational loss, respectively. By minimizing $\mathcal{L}_{CVAE}^{R}(\phi_t, \theta_t) + \mathcal{L}_{\mathrm{CVAE}}^{V}(\phi_t)$, the conditional encoder, ϕ_t, promotes $\boldsymbol{z}$ to follow the Gaussian distribution, $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, from $\boldsymbol{x}^t$ and y^t, while the conditional decoder, θ_t, reconstructs the surrogate $\hat{\boldsymbol{x}}^t$ of the original sample $\boldsymbol{x}^t$ from y^t and $\boldsymbol{z}$ (Kingma & Welling, 2014).

Knowledge Reconstruction and Feedback Consolidation. Continually training the model with Eq.A1 on the sequential tasks will incur the forgetting problem (Ye & Bors, 2021). To plastically learn new tasks while stably maintaining the previously learned tasks, we extend the intrinsic reconstruction character of CVAE to the knowledge reconstruction. We assume that the learned distribution is well retained in θ_{t-1}, and θ_t reconstructs it by knowledge reconstruction without old pseudo data.

Specifically, we freeze and store the historical decoder $p_{\theta_{t-1}}$, and assume the knowledge about the learned tasks (i.e., from task 1 to $t-1$) is retained in θ_{t-1} (Hinton et al., 2015). By inputting the current decoder p_{θ_t} and the historical decoder $p_{\theta_{t-1}}$ with the same latent variable $\boldsymbol{z}$ (Gaussian noises) and the set of labels $\boldsymbol{y}^{:t-1}$, we make their reconstruction outputs to be as consistent as possible with the following knowledge reconstruction loss,

$$\mathcal{L}_{\mathrm{CGL}}^{R}(\theta_t) = -\mathbb{E}_{p_{\theta_{t-1}}(\hat{\boldsymbol{x}}^{t-1}|y^{t-1}, \boldsymbol{z}), y \sim U(1, |\boldsymbol{y}^{:t-1}|), \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\log p_{\theta_t}(\hat{\boldsymbol{x}}^{t-1} | y, \boldsymbol{z})\right], \tag{A2}$$

where $U(\cdot)$ is discrete uniform distribution, and $|\cdot|$ is the cardinality operator. $y \sim U\left(1, \left|\boldsymbol{y}^{:t-1}\right|\right)$ means uniformly sampling y from the set of task labels learned so far. Since $p_{\theta_{t-1}}$ receives the same inputs before and after training, p_{θ_t} reconstructs the historical knowledge retained in the θ_{t-1} without data distribution drift (Bagus et al., 2022). We take the same aligning strategy for $\mathcal{L}_{LG}^{R}$ as the reconstruction loss $\mathcal{L}_{\mathrm{CVAE}}^{R}$ here. Existing strategies include single sample-based loss (e.g., L2 or L1) (Ye & Bors, 2021) and overall distribution-based loss (e.g., KL or JS) (Ramapuram et al.,

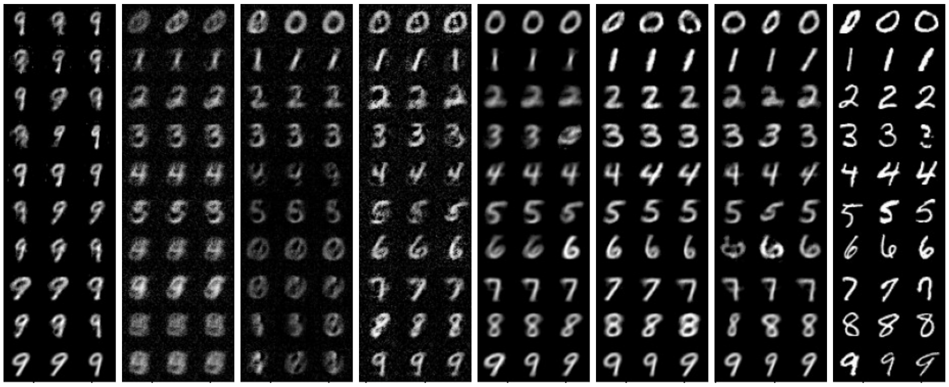

(a) Fine (b) rCGAN (c) EWC (d) MGAN (e) rCVAE (f) KFC (g) Joint (h) MNIST

Figure A1: The generated digits after the each method of (a) Fine, (b) rCGAN, (c) CEWC, (d) MGAN, (e) rCVAE, (f) KFC, (g) Joint sequentially learn (h) 10 MNIST tasks.

2020), etc. Designing aligning strategies of VAE is still an open problem (Tolstikhin et al., 2018), and we engage L2 in our experiments.

To improve the decoder's task understanding and the generating performances, we introduce a feedback consolidation strategy for KFC on the current task. This strategy is mainly inspired by GANs' discriminator-driven generation (Che et al., 2020; Verma et al., 2018), where the trained discriminator guides the generator training. We achieve feedback consolidation intuitively from two aspects: i) enhancing data consistency by aligning the posterior distribution of reconstructed samples with that of real samples; and ii) improving label-conditioned generation by integrating one additional label-encoding feedback. This mechanism is expected to assist the model in acquiring generalized knowledge of the current task, facilitating effective knowledge reconstruction to prevent forgetting when on the next task.

Formally, we decode the reconstructed data using $p_{\theta_t}(\hat{\boldsymbol{x}}^t|y^t, \boldsymbol{z})$ at first, then *re-encode* $\hat{\boldsymbol{x}}^t$ to a latent variable $\boldsymbol{z}$ by using $q_{\phi_t}(\boldsymbol{z} \mid y^t, \hat{\boldsymbol{x}}^t)$. We take $\mathrm{KL}\left[q_{\phi_t}(\boldsymbol{z} \mid y^t, \hat{\boldsymbol{x}}^t) \parallel p(\boldsymbol{z})\right]$ to design this feedback consolidation loss as,

$$\mathcal{L}^F_{\mathrm{CGL}}(\phi_t, \theta_t) = \mathbb{E}_{p_{\theta_t}(\hat{\boldsymbol{x}}^t|y^t,\boldsymbol{z})}\left[\mathrm{KL}\left[q_{\phi_t}\left(\boldsymbol{z} \mid y^t, \hat{\boldsymbol{x}}^t\right) \parallel p(\boldsymbol{z})\right]\right], \tag{A3}$$

which ensures the real data of the current task and those obtained from the decoder follow the same posterior distribution, improving generative samples' quality (Che et al., 2020).

Accompanying the reconstruction loss, variational loss with the knowledge reconstruction and feedback consolidation losses, the overall learning objective is defined as,

$$\min_{\phi_t,\theta_t}\left\{\mathcal{L}^R_{\mathrm{CVAE}}(\phi_t, \theta_t) + \mathcal{L}^V_{\mathrm{CVAE}}(\phi_t) + \lambda^r_t \mathcal{L}^R_{\mathrm{CGL}}(\theta_t) + \lambda^f_t \mathcal{L}^F_{\mathrm{CGL}}(\phi_t, \theta_t)\right\},$$

where λ^r_t and λ^f_t are trade-off parameters. We set $\lambda^r_t = t - 1$ while $\lambda^f_t = 1, t = 1, ..., T$ as the knowledge reconstruction and feedback consolidation are related to the learned $(t-1)$ tasks and the current one task, respectively (Wu et al., 2018).

B Extensive Experiments

We use the Pytorch (Paszke et al., 2019) framework with Adam optimizer (Kingma & Ba, 2015), batch-size with 64, and a learning rate of 0.005 on CIFAR10 (batch-size with 128 and learning rate of 0.0002 on MNIST and fashion MNIST). All experiments with different 5 seeds are conducted on the same Linux server with a Tesla V100-PCIe GPU.

Evaluation Metrics. We engage the training time, accuracy (ACC), and Fréchet Inception Distance (FID) as our evaluation metrics (Heusel et al., 2017). The training time is averaged over the whole T tasks. ACC is calculated on the real samples of the tasks seen so far with the evaluated classifier, which is well-trained on the generated data (Zhai et al., 2019; Yin et al., 2020). FID, instead of evaluating the generated samples directly, compares the statistics between the generated dataset and the real one. Higher ACC and lower FID indicate better generation quality.

Comparison Methods. The comparisons include the widely-used pseudo-replay methods built upon either CVAE (rCVAE) or CGAN (rCGAN) (van de Ven et al., 2020; Ramapuram et al., 2020; Ye & Bors, 2021; Shin et al., 2017), and the improved methods, CEWC Seff et al. (2017), and MGAN Liu et al. (2020); Wu et al. (2018). We also demonstrate the lower-bound and upper-bound performances of Fine-Tuning (Fine) and Joint Training (Joint), respectively. Fine-tuning conventionally trains the model without any continual learning strategy on the sequential tasks. Joint training trains the model on the combined real data of the tasks seen so far Liu et al. (2020); Wu et al. (2018).

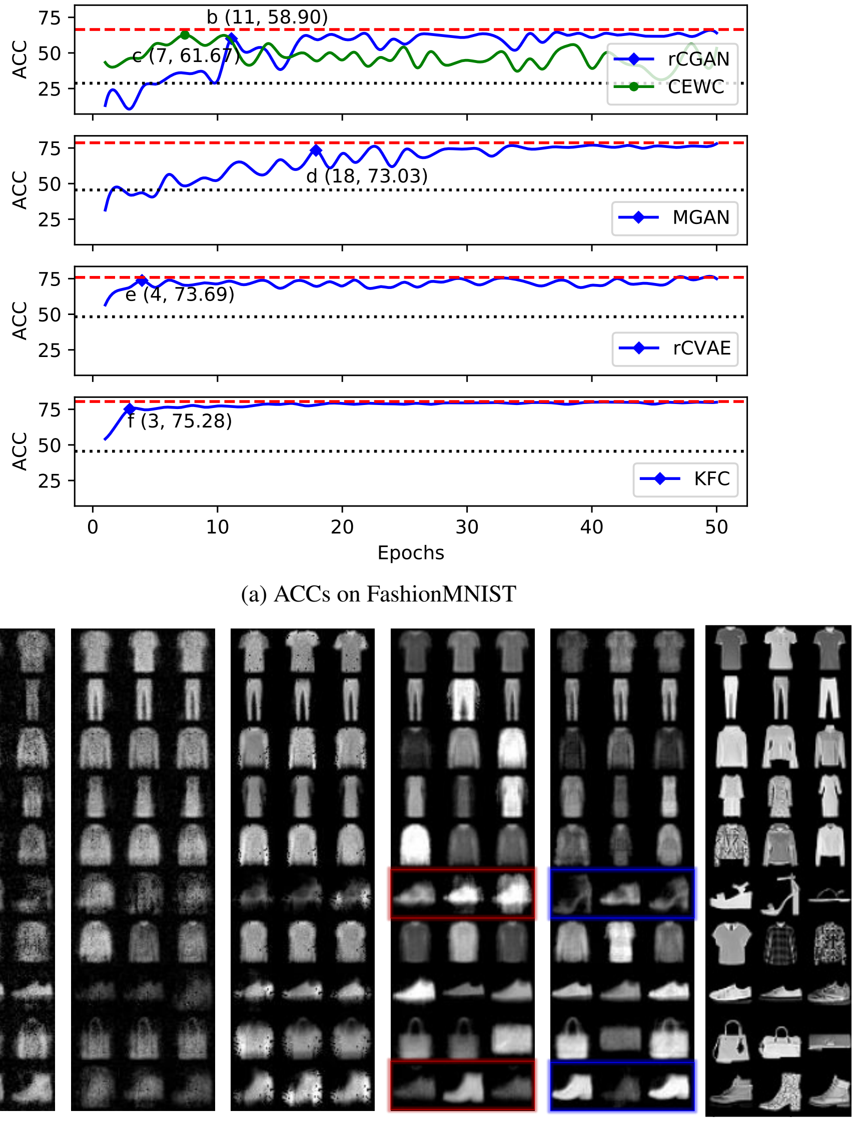
Initial Experiments on MNIST. We first confirm the forgetting problem in CGL and verify the success of KFC. We conduct experiments on continual digits generating 10 MNIST tasks (Zhai et al., 2019), where each task contains only one digital category and arrives in ascending order. For fairness, we implement all methods on the network with the same level of parameters, i.e., ≈ 0.66 million, and the same training strategy, including optimizer, epochs, learning rate, etc. After each generator learns the whole sequence of tasks, we demonstrate the generated digits from the last model. As shown in Fig.A1, Fine-Tuning incurs the forgetting problem with that it can only generate the last digit, 9. On the contrary, Joint Training could generate all learned digits, confirming that the unacceptable result of Fine-Tuning is not limited by the network capacity but the forgetting problem. Comparing all CGL methods, KFC generates the best results while CGAN-based CGL methods (i.e., rCGAN, CEWC, and MGAN) almost fail in learning long sequence tasks and inferior sample quality.

Experiments on FashionMNIST. We verify the efficiency of KFC in continually generating 10 FashionMNIST tasks. Similar to the setting on MNIST, 10 separate tasks are conducted on FashionMNIST. We calculate the ACC at different numbers of epochs (from 1 to 50). For instance, With a fixed i epoch, the model has trained i epochs on each task until sequentially finishing 10 tasks. We take an extra classifier trained on the saved model's generated samples to produce the ACC on the real test dataset. We set the lower and upper bound ACCs as the best results of Fine-Tuning and Joint Training over 50 epochs, respectively. We could verify the efficiency and effectiveness of CGL methods by comparing the number of epochs at which they reach their optimal ACC for the first time.

In Fig.A2(a), we stress out the best ACC whenever each method is first reached, along with the needed training epochs in the format of *snapshot (epochs, optimal ACC)*. Overall, KFC obtains the best ACC (75.28% versus the second-best, 73.69%, produced by rCVAE) with the least number of epochs (3 epochs versus the second-best, 4, from rCVAE) and performs the best training stability with higher sample quality. Due to the inherent training weakness (Arjovsky et al., 2017), GAN-based methods perform unstably as the number of epochs grows. rCVAE performs more consistently by its intrinsic stable training mechanism. KFC takes fewer epochs to reach the optimal ACC for the first time and achieves better training stability than rCVAE. Figs.A2(b)-(f) depicts the generated images from these snapshots, and we can find that all methods could produce the learned images when they reach their best ACC, indicating their effectiveness in alleviating forgetting problems. Notably, KFC realizes continual learning in the shortest time and achieves a satisfactory sample diversity, whereas rCVAE consumes more training epochs and confuses Sandal (the sixth class) and Ankle Boot (the last class).

Experiments on CIFAR10. We compare KFC and generative replay (implemented with rCVAE) in continually generating a complex dataset, CIFAR10 (Netzer et al., 2011). CIFAR10 is a 10-class dataset of real-world colored images, where each image is in the shape of $3 \times 32 \times 32$. We separate CIFAR10 into 4 sequential tasks according to the group of the labels:{airplane, automobile}, {bird, cat, deer}, {dog, frog}, and {horse, ship, truck} (Ye & Bors, 2021; Zhai et al., 2019). We stack the convolutional and ResNet blocks (Teoh & Rong, 2022; He et al., 2016) to build the model for KFC and generative replay methods. After the model is well-trained on 4 sequential CIFAR10 tasks, We demonstrate the generated images and record the final ACC and FID.

Fig.A3 shows the results reconstructed by the final model after it sequentially learns 4 CIFAR10 tasks. Overall, KFC achieves a better result (132.62 versus 186.17 in FID value) in a shorter training time (7h versus 12h) than the generative replay. Interestingly, KFC has, at least in visual, consistent reconstruction results for all ground truth images. In contrast, as shown in Fig.A3(b), the results of



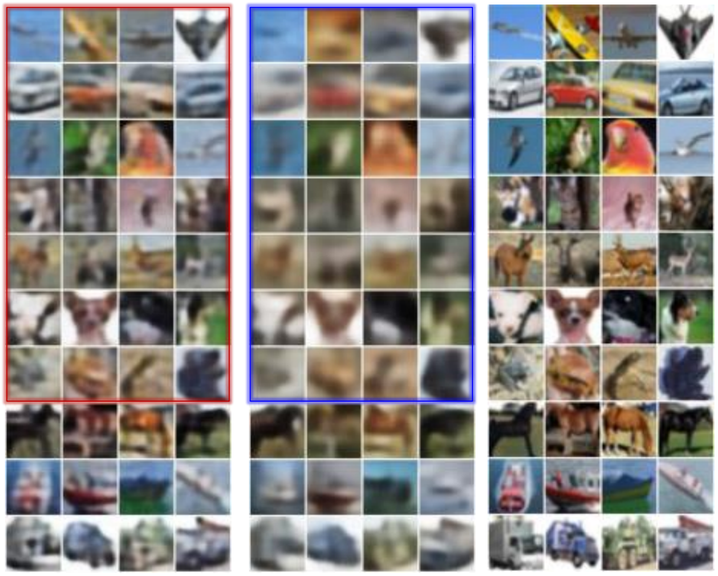
(a) ACCs on FashionMNIST

(b) rCGAN (c) CEWC (d) MGAN (e) rCVAE (f) KFC (g) FashionMNIST

Figure A2: (a) ACC after sequentially training 10 FashionMNIST tasks with different epochs. The black and red dashed lines indicate the lower and upper performances, respectively. Stressed *snapshot (epochs, optimal ACC)* indicates the number of epochs at which the method reaches its optimal ACC for the first time. (b)-(f) The generated samples of each snapshot obtained from CGL methods (b) rCGAN [700s], (c) CEWC [388s], (d) MGAN [1230s], (e) rCVAE [417s], and (f) KFC [157s] after the generator sequentially learns 10 FashionMNIST tasks in (g), where [·] is the training time in seconds to get the snapshot. The colored box highlights KFC's superior sample quality performance on previously learned tasks.

generative replay on the first 3 tasks are significantly worse than those on the last task (i.e., horse, ship, and truck in the last 3 rows), indicating that it still suffers from serious forgetting problems in learning sequential tasks.

Ablative Experiments on CIFAR10. We conducted our ablation studies on 4 CIFAR10 tasks with the comparison methods, rCVAE, rCGAN, CEWC, and MGAN, as well as two ablative baselines. They include Baseline1 which takes feedback consolidation only, and Baseline2 which takes knowledge reconstruction only built upon the same network as KFC.



(a) KFC (b) GR (c) CIFAR10

Figure A3: (a) KFC framework, and the reconstructed images of (b) KFC and (c) GR methods after the generator sequentially learns 4 CIFAR10 tasks in (d), i.e., {airplane, automobile}, {bird, cat, deer}, {dog, frog}, and {horse, ship, truck}. The colored box highlights KFC's superior sample quality performance on previously learned tasks.

Task	rCGAN		CEWC		MGAN		rCVAE		Fine-Tuning		Baseline1		Baseline2		KFC	
	ACC	FID	ACC	FID	ACC	FID	ACC	FID	ACC	FID	ACC	FID	ACC	FID	ACC	FID
1	87.95 (±2.92)	270.28	**88.45** (±9.01)	**202.65**	87.05 (±3.09)	233.43	87.35 (±0.72)	258.99	86.29 (±1.31)	271.00	86.69 (±1.33)	257.02	86.79 (±0.86)	246.32	87.95 (±0.57)	241.98
2	66.17 (±6.93)	194.66	50.05 (±2.80)	287.93	67.76 (±1.66)	204.61	66.65 (±0.59)	181.46	44.37 (±2.02)	215.40	46.96 (±2.21)	266.10	60.39 (±0.13)	195.10	**69.33** (±0.28)	**150.73**
3	55.75 (±4.49)	261.94	44.22 (±0.68)	275.06	59.92 (±0.60)	166.89	58.93 (±0.92)	176.80	38.46 (±2.24)	293.71	41.09 (±1.50)	232.41	54.42 (±0.88)	191.57	**64.45** (±0.70)	**141.74**
4	50.09 (±0.52)	237.22	38.47 (±0.91)	347.21	56.01 (±3.93)	174.95	55.11 (±0.30)	186.17	35.11 (±0.91)	214.60	36.83 (±0.96)	262.85	49.95 (±0.73)	187.46	**61.11** (±0.75)	**132.62**
T, P	$16.63h, 4.03m$		$10.42h, 4.03m$		$10.02hs, 4.01m$		$11.88h, 4.01m$		$5.98hs, 4.02m$		$6.76h, 4.02m$		$6.20h, 4.02m$		$6.89h, 4.02m$	

Table A1: ACC (%) (± std) and FID of various CGL methods evaluated on 4 sequential CIFAR-10 tasks with well trained models. T and P indicate the whole training time (in hours) and the number of network parameters (in mega), respectively.

As shown in Tab.A1, all CGL methods could prevent the generative models from forgetting to some extent compared with Fine-Tuning. This improvement is especially prominent in terms of ACC. However, GAN-based CGL methods exhibit a high standard deviation of ACC, indicating their instability similar to that in Fine-Tuning. For instance, rCGAN obtains 4.49% standard deviation (std) after learning 3 tasks, and MGAN has 3.93% std after learning 4 tasks. This high variance may be attributed to the random nature of forgetting in Fine-Tuning and the unstable training mechanisms in GAN-based methods. In contrast, rCVAE and KFC demonstrate lower variances, indicating their stable CGL capabilities. Notably, KFC outperforms others, achieving the best ACC and FID, with comparable training time, which showcases its superiority in CGL.

Besides, we could get two ablative results from Tab.A1 that 1) Baseline1 shows a similar forgetting problem as Fine-tuning. The reason is that Baseline1 is trained by only appending the CVAE loss to the feedback consolidation loss, which is responsible for increasing the generated sample quality. And 2) by expanding the intrinsic reconstruction character of CVAE to knowledge reconstruction, Baseline2 could well handle this forgetting problem and retain the historical knowledge learned so far. With the benefits from both feedback consolidation and knowledge reconstruction, KFC outperforms the widely-used pseudo-replay methods (rCVAE and rCGAN) and other improved CGL methods (CEWC and MGAN) in both time consumption and sample quality while achieving comparable classification results.