Gradient-Free Continual Learning

Grzegorz Rypeść[1], Tomasz Trzciński[1, 2]

[1]Warsaw University of Technology [2]Tooploox

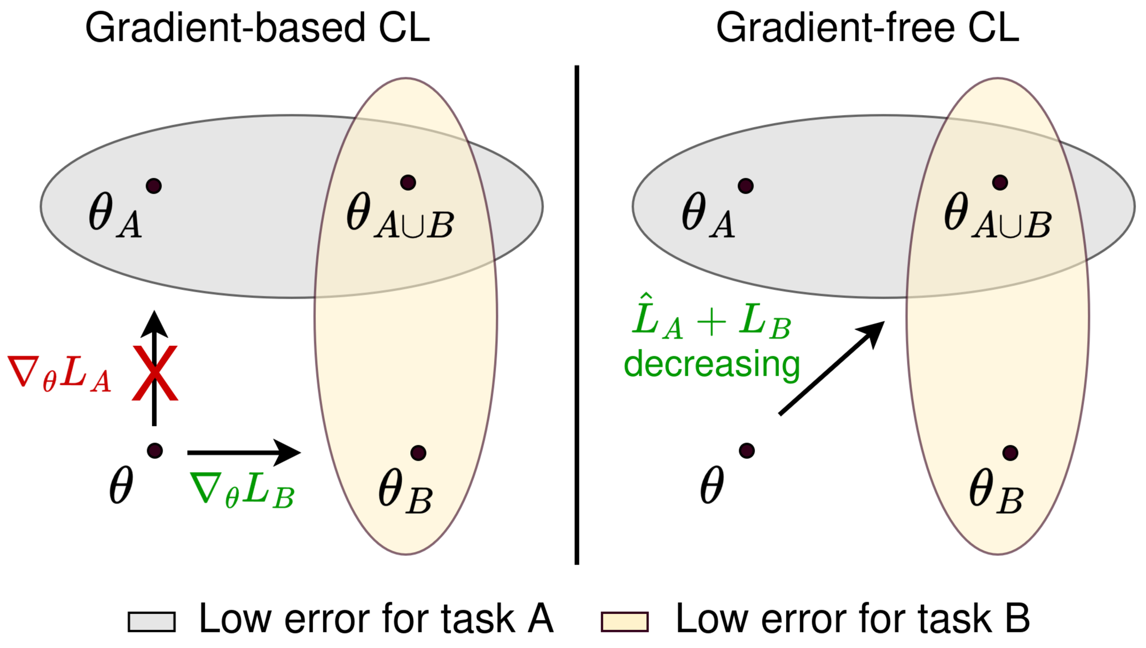
TL;DR

- We shift the focus from overcoming the lack of past data to overcoming the lack of past gradients in Continual Learning.
- We propose EvoCL, a gradient-free method that replaces backpropagation with an evolution strategy, enabling optimization even when past-task gradients are unavailable.
- We are first to jointly train both the feature extractor and a feature adaptation network.

Problem: Exemplar-free Class Incremental Learning

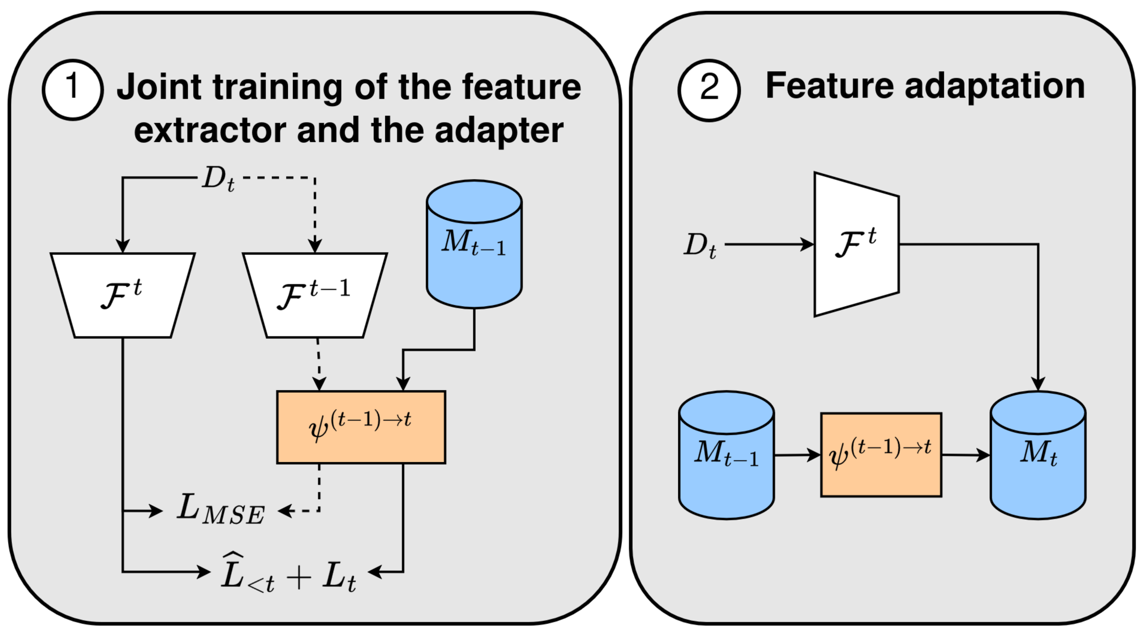
- The problem of training a model on a sequence of disjoint tasks.
- During inference we do not know the task id.
- The goal is to alleviate forgetting while allowing for high plasticity.

Motivation



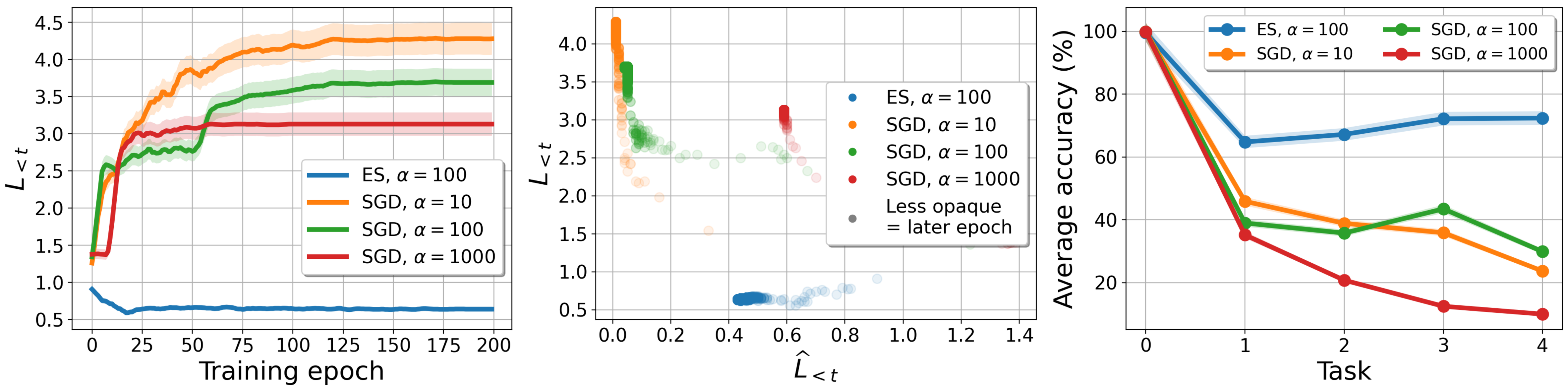
- Gradient-based optimizers fail to find a joint minimum suitable for both tasks A and B in continual learning due to the lack of gradients from task A. However, given a non-differentiable approximation of task A's loss L_A (denoted $\hat{L}_A$), a gradient-free optimizer can still identify a satisfactory solution.

Method: EvoCL

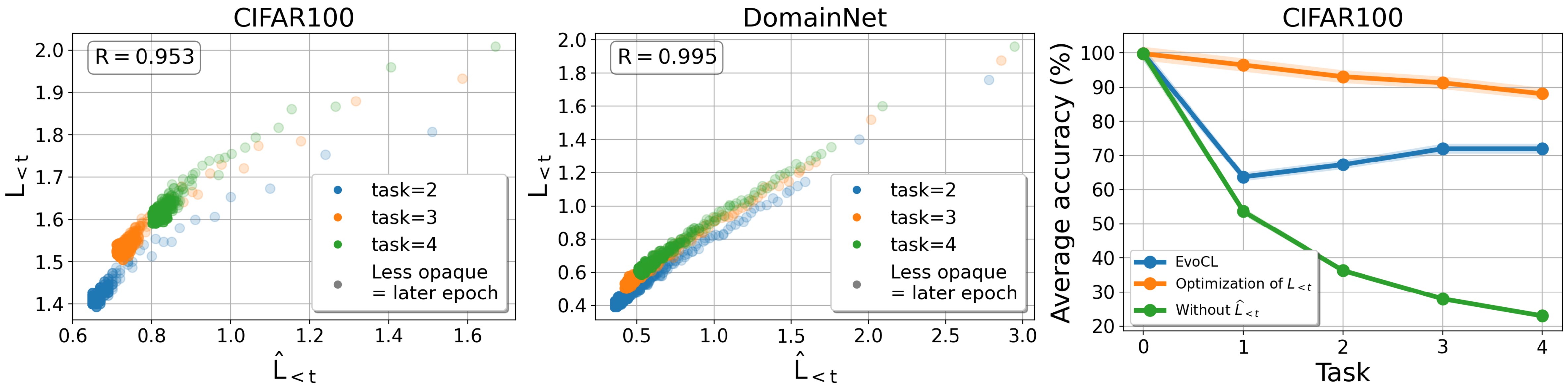


- We maintain a buffer of latent features M_t.
- We use an auxiliary MLP network (adapter $\Psi^{(t-1)\to t}$) to transform past latent features to the latent space of currently trained model.
- We train the feature extractor ($\mathcal{F}^t$) and adapter jointly using ($\mu + \lambda$) Evolution Strategy, where i-*th* individual is a pair ($\mathcal{F}^t_i$, $\psi_i^{(t-1)\to t}$).
- To create offspring we cross parents using interpolation and mutate children by adding Gaussian noise.
- We rank individuals according to loss function $L = L_t + \hat{L}_{<t} + \alpha L_{MSE}$. Only the top μ go to the next epoch, the rest is terminated :(.

Method analysis



- Comparison of ES and SGD optimizer for different α values on Fashion MNIST dataset. SGD is unable to minimize the $L_{<t}$ (left) although it minimizes the $\hat{L}_{<t}$ better than ES (middle). This leads to worse average accuracy after each task (right).



- Correlation of approximated loss for previous tasks $\hat{L}_{<t}$ and ground truth $L_{<t}$ (left, middle) after second, third and fourth task. Each point presents the loss values after a training epoch. We can see that losses are correlated and minimizing $\hat{L}_{<t}$ decreases $L_{<t}$. EvoCL performs poorly without $\hat{L}_{<t}$ in the loss function (right), and much better with $L_{<t}$ which is the upper-bound assuming we have access to old data.

Results

Method	DomainNet						FGVCAircraft					
	T=6		T=12		T=18		T=5		T=10		T=20	
	$A_{last}\uparrow$	$A_{inc}\uparrow$	$A_{last}\uparrow$	$A_{inc}\uparrow$	$A_{last}\uparrow$	$A_{inc}\uparrow$	$A_{last}\uparrow$	$A_{inc}\uparrow$	$A_{last}\uparrow$	$A_{inc}\uparrow$	$A_{last}\uparrow$	$A_{inc}\uparrow$
Upper bound	82.3±0.1						87.0±0.1					
Finetune	21.6±0.3	38.2±0.2	15.8±0.2	32.6±0.2	12.3±0.1	27.2±0.2	24.3±0.1	44.0±0.1	14.3±0.1	34.5±0.2	10.9±0.2	27.9±0.1
LwF	54.3±0.1	67.7±0.1	40.4±0.1	54.1±0.1	37.4±0.1	45.7±0.1	30.0±0.1	44.2±0.1	28.0±0.2	46.5±0.1	14.7±0.1	30.5±0.1
EWC	21.6±0.1	38.2±0.2	15.8±0.1	32.6±0.1	12.3±0.1	27.2±0.1	24.3±0.1	44.0±0.1	14.3±0.1	34.5±0.2	10.9±0.2	27.9±0.1
PASS	44.5±0.3	58.2±0.4	27.0±0.3	42.3±0.2	18.1±0.3	36.9±0.2	26.3±0.2	42.7±0.1	26.4±0.1	41.0±0.1	18.9±0.3	28.2±0.2
IL2A	46.9±0.4	61.3±0.3	29.4±0.1	45.5±0.2	20.8±0.1	35.1±0.1	28.4±0.1	38.1±0.1	25.3±0.1	39.1±0.1	18.2±0.1	28.7±0.1
FeTrIL	68.9±0.2	73.2±0.2	54.9±0.3	61.2±0.2	38.6±0.3	49.3±0.2	31.0±0.3	45.5±0.2	30.5±0.2	48.4±0.2	30.5±0.3	43.3±0.2
FeCAM	71.5±0.1	78.4±0.1	56.2±0.2	66.9±0.2	40.2±0.1	50.9±0.1	33.3±0.2	47.0±0.2	32.9±0.2	50.2±0.2	31.0±0.1	50.0±0.2
DS-AL	70.4±0.2	77.9±0.2	55.8±0.2	64.1±0.2	41.4±0.2	53.8±0.2	32.6±0.3	46.7±0.2	32.3±0.2	51.4±0.2	31.2±0.2	48.7±0.2
EvoCL	**73.1**±0.4	**81.6**±0.3	**63.0**±0.4	**72.3**±0.3	**43.8**±0.3	**57.2±0.1**	**34.8**±0.3	**48.2**±0.3	**34.6**±0.3	**52.4**±0.3	**32.1**±0.4	**52.7**±0.2

- Average incremental and last accuracy in EFCIL fine-grained scenarios when utilizing a pre-trained feature extractor (ViT-small). We add a learnable MLP network before the last transformer block and keep the rest of the model frozen. We report the mean and standard deviation of three runs. EvoCL achieves better performance than counterparts.

Method	MNIST				FashionMNIST				CIFAR100			
	T=3		T=5		T=3		T=5		T=5		T=10	
	$A_{last}\uparrow$	$A_{inc}\uparrow$	$A_{last}\uparrow$	$A_{inc}\uparrow$	$A_{last}\uparrow$	$A_{inc}\uparrow$	$A_{last}\uparrow$	$A_{inc}\uparrow$	$A_{last}\uparrow$	$A_{inc}\uparrow$	$A_{last}\uparrow$	$A_{inc}\uparrow$
Upper bound	98.7±0.1				92.1±0.1				71.8±0.2			
Finetune	40.2±0.2	65.9±0.2	21.6±0.2	52.8±0.2	21.1±0.2	54.8±0.2	27.9±0.2	40.2±0.2	14.7±0.2	17.1±0.2	12.6±0.1	14.0±0.2
LwF	85.6±0.1	91.3±0.1	47.4±0.2	72.6±0.2	27.0±0.1	57.9±0.2	32.1±0.1	50.9±0.1	22.1±0.2	34.1±0.2	19.1±0.1	31.1±0.1
EWC	76.7±0.2	82.5±0.1	44.8±0.2	69.2±0.2	26.5±0.2	54.3±0.2	30.0±0.2	48.8±0.1	20.9±0.1	30.3±0.1	17.4±0.2	28.6±0.2
PASS	53.7±0.3	69.8±0.2	44.8±0.2	55.7±0.2	22.7±0.1	56.9±0.1	28.5±0.3	42.8±0.2	18.1±0.1	29.0±0.1	17.3±0.3	30.9±0.2
IL2A	56.9±0.1	61.3±0.1	29.4±0.1	45.5±0.1	20.8±0.1	35.1±0.1	28.4±0.1	43.1±0.1	19.3±0.1	31.1±0.1	14.2±0.1	28.7±0.1
FeTrIL	78.3±0.1	80.9±0.1	75.5±0.2	79.9±0.2	61.6±0.2	68.1±0.1	59.5±0.2	62.0±0.2	21.7±0.1	34.1±0.2	18.8±0.1	30.5±0.1
FeCAM	79.4±0.1	82.5±0.1	76.2±0.1	81.4±0.1	64.0±0.3	70.3±0.2	63.8±0.1	67.1±0.2	22.2±0.1	34.7±0.1	20.6±0.2	32.9±0.2
DS-AL	84.2±0.2	87.8±0.2	75.4±0.2	77.0±0.2	73.6±0.1	75.3±0.2	68.1±0.2	64.1±0.2	21.7±0.1	35.6±0.2	20.0±0.2	32.6±0.2
EvoCL	**92.3**±0.3	**95.8**±0.2	**81.8**±0.2	**89.9**±0.2	**77.2**±0.4	**78.0**±0.2	**72.0**±0.2	**74.9**±0.2	**24.8**±0.2	**37.2**±0.1	**21.3**±0.1	**35.8**±0.1

- Results when training from scratch. This is a much harder scenario than finetuning the model, as the model sees much less data and starts with random weights. Despite this, EvoCL also performs better than baselines. As the F_t we use a 3-layered MLP network and train all parameters.

Find out more!

- https://github.com/grypesc/EvoCL
- https://www.linkedin.com/in/grypesc
- I am looking for ambitious people interested in quantitative finances!

Contact: grzegorz.rypesc.dokt@pw.edu.pl