Algorithm 1 Meta-Learning with Separate Data Distillation

Require: Tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K$; number of per-task examples M; initialization distribution $p(\theta_0)$
  for i in $[K]$ tasks **do**
    $\tilde{X}_i \leftarrow \text{DatasetDistillation}(\tilde{X}_i, \mathcal{T}_i, M)$
  end for
  Initialize: Network f; $\theta_0 \sim p(\theta_0)$
  $f^* \leftarrow \text{Optimizer}(\text{network: } f, \text{ data: } \{\tilde{X}_i\}_{i=1}^k)$;
  return f^*

Algorithm 2 Meta-Learning with Mixed Dataset Distillation (Reptile-like)

Require: Tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K$; number of distilled exmples M; meta-training iterations T; init. distribution $p(\theta_0)$; number of initializations N; learning rate ϵ.
  Initialize: $\tilde{X}_0 = \{\tilde{x}_i\}_{i=1}^M$;
  for t in $[T]$ iterations **do**
    for k in $[K]$ tasks **do**
      $\tilde{X}_t^{(k)} \leftarrow \text{DatasetDistillation}(\tilde{X}_t, \mathcal{T}_k, M, p(\theta_0))$
      $\tilde{X}_t \leftarrow \tilde{X}_t + \epsilon(\tilde{X}_t^{(k)} - \tilde{X}_t)$
    end for
  end for
  Initialize: Network f; $\theta_0 \sim p(\theta_0)$;
  $f^* \leftarrow \text{Optimizer}(\text{network: } f_{\theta_0}, \text{ data: } \tilde{X}_T)$;
  return f^*

Algorithm 3 Meta-Learning with Mixed Dataset Distillation (MAML-like)

Require: Tasks $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_K$; number of distilled exmples M; meta-training iterations T; initialization distribution $p(\theta)$; number of initializations N; inner-loop learning rate α; meta-training learning rate β; synthetic loss $\mathcal{L}$.

Initialize: $\tilde{X}_0 = \{\tilde{x}_i\}_{i=1}^{M}$;
for t in $[T]$ iterations **do**
  for n in $[N]$ initializations **do**
    Sample $\theta_0^{(n)} \sim p(\theta_0)$
    $\theta_1^{(n)} \leftarrow \theta_0^{(n)} - \alpha \nabla_\theta \mathcal{L}(\theta_0^{(n)}, \tilde{X}_t)$
    $\tilde{X}_t \leftarrow \tilde{X}_t - \beta \frac{1}{K} \sum_{k=1}^{K} \nabla_x \mathcal{L}_k(\theta_1^{(n)}, \mathcal{T}_k)$
  end for
end for
Initialize: Network f; $\theta_0 \sim p(\theta_0)$;
$f^* \leftarrow$ Optimizer(network: f_{θ_0}, data: $\tilde{X}_T$);
return f^*

Algorithm 4 Meta-Learning with Mixed Dataset Distillation (MAML-like)

Require: Tasks $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_K$; number of distilled exmples M; meta-training iterations T; initialization distribution $p(\theta)$; number of initializations N; inner-loop learning rate α; meta-training learning rate β; synthetic loss $\mathcal{L}$.

Initialize: $\tilde{X} = \{\tilde{x}_i\}_{i=1}^{M}$;
for t in $[T]$ iterations **do**
  for n in $[N]$ initializations **do**
    Sample $\theta_0^{(n)} \sim p(\theta_0)$
    $\theta_1^{(n)} \leftarrow \theta_0^{(n)} - \alpha \nabla_\theta \mathcal{L}(\theta_0^{(n)}, \tilde{X}_t)$
  end for
  $\tilde{X}_t \leftarrow \tilde{X}_t - \beta \frac{1}{NK} \sum_{k=1}^{K} \sum_{n=1}^{N} \nabla_x \mathcal{L}(\theta_1^{(n)}, \mathcal{T}_k)$
end for
Initialize: Network f; $\theta_0 \sim p(\theta_0)$;
$f^* \leftarrow$ Optimizer(network: f_{θ_0}, data: $\tilde{X}_T$);
return f^*