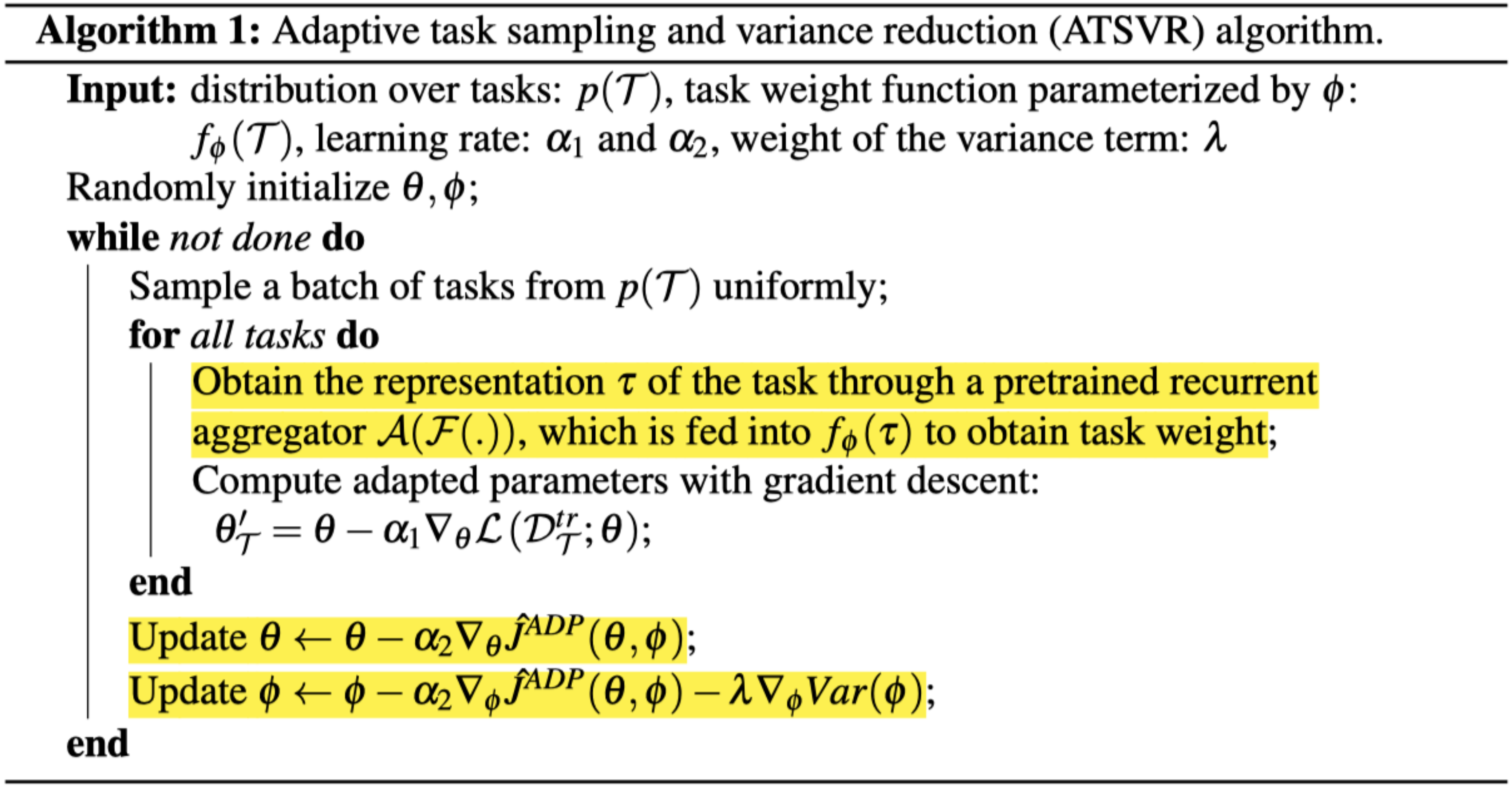
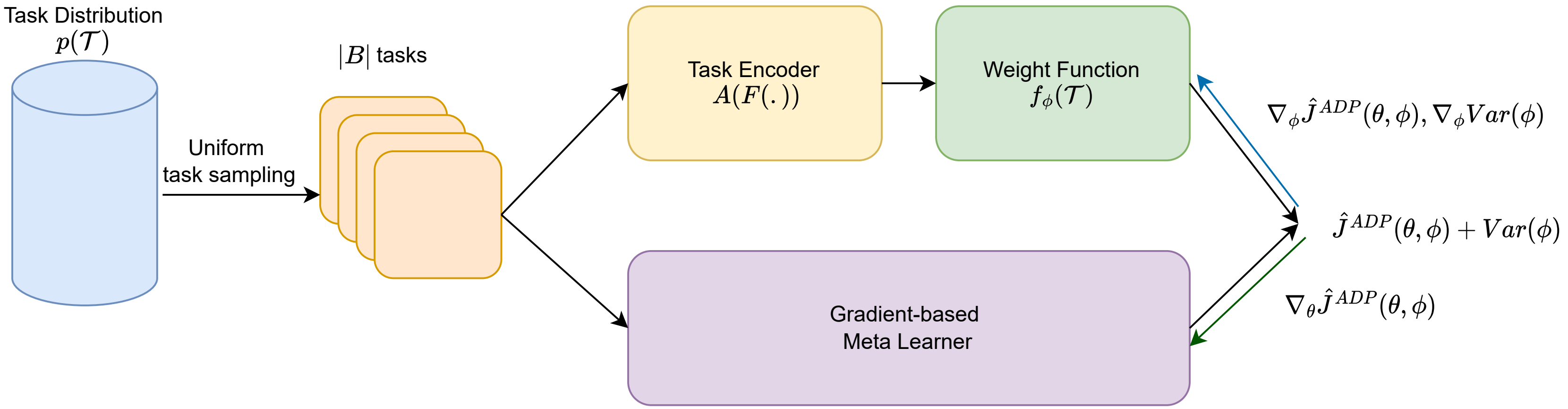


Algorithm 1: Adaptive task sampling and variance reduction (ATSVR) algorithm.

```
Input: distribution over tasks: $p(\mathcal{T})$, task weight function parameterized by $\phi$:
        $f_\phi(\mathcal{T})$, learning rate: $\alpha_1$ and $\alpha_2$, weight of the variance term: $\lambda$
Randomly initialize $\theta, \phi$;
while not done do
    Sample a batch of tasks from $p(\mathcal{T})$ uniformly;
    for all tasks do
        Obtain the representation $\tau$ of the task through a pretrained recurrent
        aggregator $\mathcal{A}(\mathcal{F}(.))$, which is fed into $f_\phi(\tau)$ to obtain task weight;
        Compute adapted parameters with gradient descent:
          $\theta'_\mathcal{T} = \theta - \alpha_1 \nabla_\theta \mathcal{L}(\mathcal{D}^{tr}_\mathcal{T}; \theta)$;
    end
    Update $\theta \leftarrow \theta - \alpha_2 \nabla_\theta \hat{J}^{ADP}(\theta, \phi)$;
    Update $\phi \leftarrow \phi - \alpha_2 \nabla_\phi \hat{J}^{ADP}(\theta, \phi) - \lambda \nabla_\phi Var(\phi)$;
end
```


Task Distribution
$p(\mathcal{T})$
Uniform
task sampling
$|B|$ tasks
Task Encoder
$A(F(.))$
Weight Function
$f_\phi(\mathcal{T})$
$\nabla_\phi \hat{J}^{ADP}(\theta, \phi), \nabla_\phi Var(\phi)$
$\hat{J}^{ADP}(\theta, \phi) + Var(\phi)$
$\nabla_\theta \hat{J}^{ADP}(\theta, \phi)$
Gradient-based
Meta Learner