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

**Input:** distribution over tasks: $p(T)$, task weight function parameterized by $\phi$: $f_\phi(T)$, 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 $p(T)$ uniformly;

for all tasks do

Obtain the representation $\tau$ of the task through a pretrained recurrent aggregator $A(F(.))$, which is fed into $f_\phi(\tau)$ to obtain task weight;

Compute adapted parameters with gradient descent:

\[
\theta'_{\tau} = \theta - \alpha_1 \nabla_\theta \mathcal{L}(D^{tr}_{\tau}; \theta);
\]

end

Update $\theta \leftarrow \theta - \alpha_2 \nabla_\theta \hat{J}^{ADP}(\theta, \phi)$;

Update $\phi \leftarrow \phi - \alpha_2 \nabla_\phi \hat{J}^{ADP}(\theta, \phi) - \lambda \nabla_\phi \text{Var}(\phi)$;

end