Algorithm 1 The ADAM algorithm computes a batch stochastic gradient to compute momentum and RMSProp vectors at each timestep, with a bias correction step accounting for first and second moment estimates. Model parameters are updated using this recursive gradient information.

Require: parameters θ, stochastic objective function $f_i(\theta)$
Require: learning rate η, exponential decay rates $\beta_1, \beta_2 \in [0, 1)$, tolerance ϵ, batch size
Initialize: initial parameter vector θ_0, initial 1^{st} moment vector $m_0 = 0$, initial 2^{nd} moment vector $v_0 = 0$, initial timestep $t = 0$

while θ_t is not converged **do**
  $t = t + 1$
  $g_t = \nabla_\theta f_t(\theta_{t-1})$ (*batch gradient at iteration* t)
  $m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (*[Monentum] udpate baised first moment estimate*)
  $v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (*[RMSProp] udpate baised second raw moment estimate*)
  $\hat{m}_t = m_t/(1 - \beta_1^t)$ (*[Monentum] bias-corrected first moment estimate*)
  $\hat{v}_t = v_t/(1 - \beta_2^t)$ (*[RMSProp] bias-corrected second raw moment estimate*)
  $\theta_t = \theta_{t-1} - \eta \cdot \hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon)$ (*[Momentum + RMSProp] update parameters*)
end while
return θ_t

Algorithm 2 SARAH +

Require: objective function $f(\theta)$
Require: learning rate $\alpha > 0$, inner loop size m, outer loop size T
Initialize: initialize arbitrary parameter vector $\tilde{\theta}_0 \in \mathbb{R}$
Iterate:

```
for s = 1, 2, ..., T do
    θ_0 = θ̃_{s−1}
    v_0 = (1/n) Σ_{i=1}^{n} ∇f_i(θ_0)      (outer loop full gradient computation)
    θ_1 = θ_0 − α v_0        (compute one outer loop parameter update for inner loop)
    t = 1
    while ||v_{t−1}||^2 > γ||v_0||^2 and t < m do
        Sample i ∈ {1, ..., n} uniformly at random
        v_t = ∇f_{i_t}(θ_t) − ∇f_{i_t}(θ_{t−1}) + v_{t−1}      (gradient estimate (SARAH update))
        θ_{t+1} = θ_t − α v_t      (inner loop parameter update)
        t = t+1
    end for
    Set θ̃_s = θ_t
end for
```

In LaTeX notation:

- **for** $s = 1, 2, ..., T$ **do**
  - $\theta_0 = \tilde{\theta}_{s-1}$
  - $v_0 = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\theta_0)$ (*outer loop full gradient computation*)
  - $\theta_1 = \theta_0 - \alpha v_0$ (*compute one outer loop parameter update for inner loop*)
  - $t = 1$
  - **while** $||v_{t-1}||^2 > \gamma||v_0||^2$ **and** $t < m$ **do**
    - Sample $i \in \{1, ..., n\}$ uniformly at random
    - $v_t = \nabla f_{i_t}(\theta_t) - \nabla f_{i_t}(\theta_{t-1}) + v_{t-1}$ (*gradient estimate (SARAH update)*)
    - $\theta_{t+1} = \theta_t - \alpha v_t$ (*inner loop parameter update*)
    - t = t+1
  - **end for**
  - Set $\tilde{\theta}_s = \theta_t$
- **end for**

Algorithm 3 SVRG

Require: objective function $f(\theta)$
Require: learning rate $\alpha > 0$, inner loop size m, outer loop size T
Initialize: initialize arbitrary parameter vector $\tilde{\theta}_0 \in \mathbb{R}$
Iterate:
  for $s = 1, 2, ..., T$ **do**
    $\tilde{\theta} = \tilde{\theta}_{s-1}$
    $\tilde{g} = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{\theta})$ (*outer loop full gradient computation*)
    $\theta_0 = \tilde{\theta}$
    for $t = 1, ..., m$ **do**
      Sample $i \in \{1, ..., n\}$ uniformly at random
      $v_{t-1} = \nabla f_{i_t}(\theta_{t-1}) - \nabla f_{i_t}(\tilde{\theta}) + \tilde{g}$ (*inner loop gradient approximation*)
      $\theta_t = \theta_{t-1} - \alpha \cdot v_{t-1}$ (*inner loop parameter update*)
    end for
    Option 1: Set $\tilde{\theta}_s = \theta_t$ with $t \in \{0, 1, ..., m\}$ chosen uniformly at random
    Option 2: Set $\tilde{\theta}_s = \theta_m$
  end for

Algorithm 4 The SARAH algorithm is identical to SVRG except for the *SARAH update*, which modifies the stochastic gradient estimate to use recursive gradient estimate information rather than the initialized gradient to update the gradient estimate in the inner loop.

Require: objective function $f(\theta)$
Require: learning rate $\alpha > 0$, inner loop size m, outer loop size T
Initialize: initialize arbitrary parameter vector $\tilde{\theta}_0 \in \mathbb{R}$
Iterate:
for $s = 1, 2, ..., T$ **do**
  $\theta_0 = \tilde{\theta}_{s-1}$
  $v_0 = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\theta_0)$ (*outer loop full gradient computation*)
  $\theta_1 = \theta_0 - \alpha v_0$ (*compute one outer loop parameter update for inner loop*)
  for $t = 1, ..., m-1$ **do**
    Sample $i \in \{1, ..., n\}$ uniformly at random
    $v_t = \nabla f_{i_t}(\theta_t) - \nabla f_{i_t}(\theta_{t-1}) + v_{t-1}$ (*gradient estimate (SARAH update)*)
    $\theta_{t+1} = \theta_t - \alpha v_t$ (*inner loop parameter update*)
  end for
  Set $\tilde{\theta}_s = \theta_t$ with t chosen uniformly at random from $\{0, 1, ..., m\}$
end for