AME: Attention and Memory Enhancement in Hyper-Parameter Optimization Supplementary Material

A. Preliminary Knowledge

AME employs ASHA as the trial scheduler by default, but it is also able to be combined with Hyperband (*see* Tab. 3). Next, we will briefly introduce these schedulers, including SHA [23, 25], ASHA [32] and Hyperband [31].

Successive Halving Algorithm (SHA). The central idea of SHA (*see* Alg. I) is to compare the performance of n trials set by different hyper-parameter configurations under finite total budget B in each rung t, discarding the poorly performing trials and leaving n_t/η high-performance ones. The time interval r_t between two rungs increases exponentially. After several repetitions, only the best configuration h^* is left. Since low-performance trials are stopped early, the actual budget consumed by each trial is unequal. The sum of the actual budget of all n trials should be equal to the total budget B.

Asynchronous SHA (ASHA). ASHA is an asynchronous parallel SHA. The selection of candidates for the next rung is performed while the training or evaluation of other networks in current rung is in progress. Intuitively, as long as the evaluation indicator of one trial in a certain rung is greater than the corresponding threshold, ASHA will promote it to the next rung for training instead of waiting for the evaluation in the whole rung to be completed. In addition, the thresholds of each rung are recalculated based on the newly obtained evaluation indicators. Note that the RunReturnValLoss() subroutine (*see* Alg. I) in ASHA is asynchronous, which accelerates the speed of SHA.

Hyperband. Hyperband (*see* Alg. II) aims to make a trade-off between n (exploration) and B/n (exploitation) by repeatedly calling SHA. In the early stage, the scheduler needs to explore as many new configurations as possible; in the later stage, it gradually focuses on high-performance trials. Note that the later trials are selected from the previous ones. Hyperband is a two-layer loop, one layer is to choose different combinations of (n, r), and the other is to perform SHA for each combination. In each loop of different combinations of (n, r), configurations to be evaluated decrease and the budget for each configuration increases.

Algorithm I SHA / ASHA

Input: Maximum Budget R, Minimum Budget r, Maximum Number of Configurations n, Reduction Factor η, Minimum Early-Stopping Rate s, Search Space $\mathcal{H}$.
Output: Best Configuration h^*

$s_{\max} = \lfloor \log_\eta (R/r) \rfloor$,
$H = \text{CONFIGURATIONSAMPLING}(n, \mathcal{H})$
for $t \in \{0, 1, \cdots, s_{\max} - s\}$ **do** *// All configurations trained for a given t constitute a rung.*
  $n_t = \lfloor n\eta^{-t} \rfloor$, $r_t = r\eta^{t+s}$
  $L = \text{RUNRETURNVALLOSS}(h, r_t) : h \in H$
  $H = \text{GETTOPK}(H, L, n_t/\eta)$
end for
return Best Configuration $h^* \in H$

Algorithm II Hyperband

Input: Maximum Budget R, Minimum Budget r, Maximum Number of Configurations $n_{\max}$, Reduction Factor η, Search Space $\mathcal{H}$.
Output: Best Configuration h^*

$s_{\max} = \lfloor \log_\eta n_{\max} \rfloor$, $t_{\max} = \lfloor \log_\eta (R/r) \rfloor$,
$s_0 = t_{\max} - s_{\max}$
$H = \text{CONFIGURATIONSAMPLING}(n_{\max}, \mathcal{H})$
for $s \in \{s_{\max}, s_{\max} - 1, \cdots, 0\}$ **do**
  $n = \lfloor \frac{s_{\max}+1}{s+1} \eta^s \rfloor$, $r = R\eta^{-(s+s_0)}$
  $H_s = \text{CONFIGURATIONSAMPLING}(n, H)$
  for $t \in \{0, 1, \cdots, s + s_0\}$ **do** *// Call SHA.*
    $n_t = \lfloor n\eta^{-t} \rfloor$, $r_t = r\eta^t$
    $L = \text{RUNRETURNVALLOSS}(h, r_t) : h \in H_s$
    $H_s = \text{GETTOPK}(H_s, L, n_t/\eta)$
  end for
end for
return Best Configuration $h^* \in H$

B. Experimental Details

The supplementary results are shown in Fig. I, II, III, IV. The complete AME algorithm is shown in Alg. III. Examples of searched configurations are shown in Tab. I.

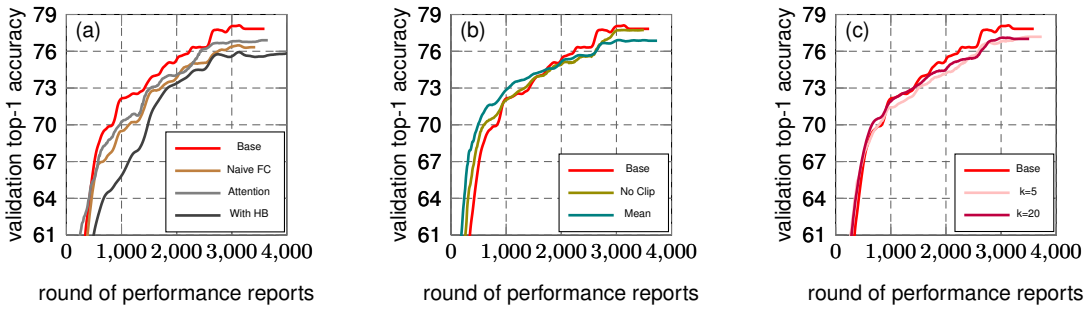

Figure III. Performance comparison in ablation studies of AME (C100). (a) Results in Tab. 3. (b) Results in Tab. 5. (c) Results in Tab. 6. Base: AME algorithm with default settings, described in Sec. 4.1. *Carry on to* Fig. 5 *in the main text.*

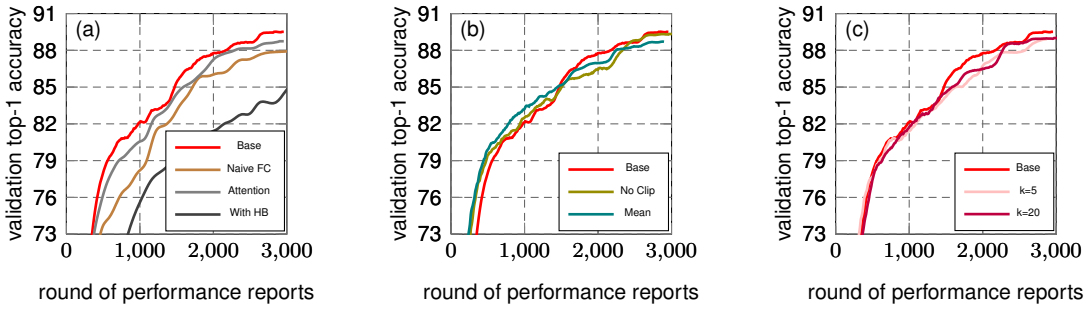

Figure IV. Performance comparison in ablation studies of AME (Cars). (a) Results in Tab. 3. (b) Results in Tab. 5. (c) Results in Tab. 6. Base: AME algorithm with default settings, described in Sec. 4.1. *Carry on to* Fig. 5 *in the main text.*

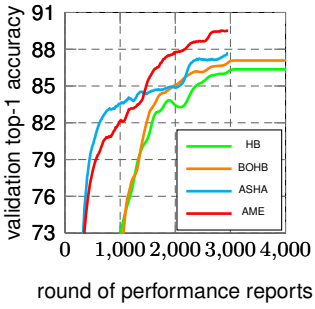

Figure I. Performance comparison of four algorithms in Standford Cars. *Carry on to* Fig. 3 *in the main text.*

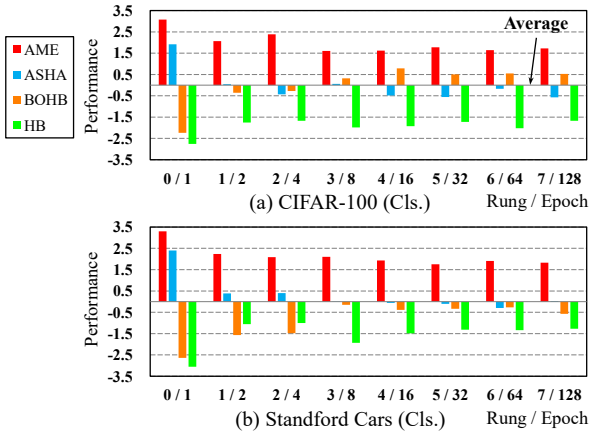

Figure II. Performance comparison of four algorithms in each rung. The bar charts represent the difference of each method relative to the average. *Carry on to* Fig. 4 *in the main text.*

Algorithm III Attention and Memory Enhancement (AME)

Input: Configuration Search Space $\mathcal{H}$, Evaluated Configuration Set $\mathcal{H}_E$, Unevaluated Configuration Set $\mathcal{H}_U$, Trial (Network) f_h with Hyper-parameter Configuration h, Configuration with Evaluation Indicator $\hat{h}$.
Output: Best Configuration h^*

function TRIALSCHEDULER(f_h) *// Queue for trials in running Q_R, Queue for trials in termination Q_T.*
  if Start a new trial **then**
    CONFSEARCHER(Need a new configuration)
    Get h from $\mathcal{H}_U$. Move f_h to Q_R.
  else if f_h is evaluated with good performance **then**
    Add $\hat{h}$ in $\mathcal{H}_E$. Move f_h to Q_R.
    CONFSEARCHER(Get an evaluated conf.)
  else if f_h is evaluated with bad performance **then**
    Add $\hat{h}$ in $\mathcal{H}_E$. Move f_h to Q_T.
    CONFSEARCHER(Get an evaluated conf.)
  end if
end function
function CONFSEARCHER($\cdot$) *// Reward $\mathcal{R}$, Action $\mathcal{A}$, State $\mathcal{S}$, The number of input conf. k, A constant ρ.*
  if Need a new configuration **then**
    if $|\mathcal{H}_E| \leq \rho k$ $(\rho \geq 1)$ **then**
      Randomly sample h from $\mathcal{H}$. Add h to $\mathcal{H}_U$.
    else
      Randomly sample $\hat{h_1}, \hat{h_2}, \cdots, \hat{h_k}$ from $\mathcal{H}_E$.
      $h = g_{ag}(\{\hat{h_1}, \hat{h_2}, \cdots, \hat{h_k}\})$. Add h to $\mathcal{H}_U$.
    end if
  else if Get an evaluated conf. & $|\mathcal{H}_E| > \rho k$ **then**
    Randomly sample $\hat{h_0}, \hat{h_1}, \cdots, \hat{h_k}$ from $\mathcal{H}_E$.
    Calculate $\mathcal{R}$ with h_0 as $\mathcal{A}$, $\hat{h_1}, \hat{h_2}, \cdots, \hat{h_k}$ as $\mathcal{S}$.
    Training $Agent$ with loss $\mathcal{L}_{ag}$.
  end if
end function
// The traversal order is roughly summarized:
for Every trial f_h in each rung **do**
  TRIALSCHEDULER(f_h)
end for
return Best Configuration $h^* \in \mathcal{H}_E$.

Task	Methods	Head	Backbone	Optimizer	LR	WD	BS
CLS (C10)	PBT	-	-	-	0.043	6.8e-4	-
	PB2	-	-	-	0.008	4.6e-4	-
	BayesOpt	-	-	-	0.016	1.6e-4	-
	Dragonfly	-	-	-	0.043	2.7e-4	-
	ZOOpt	-	-	-	0.073	8.0e-5	-
	Hyperband	-	ResNet34	Adamax	0.001	5.0e-4	52
	BOHB	-	ResNet18	Adam	0.001	0	28
	ASHA	-	ResNet34	Adamax	0.005	0	32
	AME(Ours)	-	ResNet34	SGD	0.045	5.0e-4	16
CLS (C100)	PBT	-	-	-	0.072	2.2e-4	-
	PB2	-	-	-	0.017	6.6e-4	-
	BayesOpt	-	-	-	0.037	1.2e-4	-
	Dragonfly	-	-	-	0.015	9.5e-4	-
	ZOOpt	-	-	-	0.013	4.3e-4	-
	Hyperband	-	ResNet50	Adamax	0.075	1.5e-4	20
	BOHB	-	ResNet34	Adadelta	0.1	3.0e-4	12
	ASHA	-	ResNet34	SGD	0.075	1.5e-4	20
	AME(Ours)	-	ResNet34	SGD	0.035	1.0e-5	12
CLS (Cars)	PBT	-	-	-	0.049	5.0e-5	-
	PB2	-	-	-	0.009	4.8e-4	-
	BayesOpt	-	-	-	0.013	5.8e-4	-
	Dragonfly	-	-	-	0.019	9.2e-4	-
	ZOOpt	-	-	-	0.017	4.4e-5	-
	Hyperband	-	ResNet34	Adagrad	0.025	3.5e-4	12
	BOHB	-	ResNet18	SGD	0.095	9.0e-4	24
	ASHA	-	ResNet18	SGD	0.075	0	32
	AME(Ours)	-	ResNet34	SGD	0.02	1.0e-5	12
DET (VOC)	PBT	-	-	-	0.012	1.1e-4	-
	PB2	-	-	-	0.007	8.2e-4	-
	BayesOpt	-	-	-	0.015	7.6e-4	-
	Dragonfly	-	-	-	0.014	2.6e-4	-
	ZOOpt	-	-	-	0.019	3.1e-5	-
	Hyperband	RepPoints	ResNeSt50	SGD	0.004	3.0e-4	4
	BOHB	CascadeRCNN	ResNeSt50	Adadelta	0.014	5.0e-5	12
	ASHA	RepPoints	ResNeSt50	Adadelta	0.007	3.5e-4	8
	AME(Ours)	RetineNet	ResNeSt50	SGD	0.007	1.0e-5	8
SEG (VOC)	PBT	-	-	-	0.014	5.7e-4	-
	PB2	-	-	-	0.021	8.8e-4	-
	BayesOpt	-	-	-	0.019	5.2e-5	-
	Dragonfly	-	-	-	0.012	1.8e-4	-
	ZOOpt	-	-	-	0.013	2.2e-5	-
	Hyperband	PSPNet	ResNet50	Adadelta	0.02	2.0e-4	12
	BOHB	DANet	ResNeSt50	Adadelta	0.005	9.0e-4	12
	ASHA	PSANet	ResNeXt50	SGD	0.018	9.0e-4	12
	AME(Ours)	DANet	ResNeXt50	Adadelta	0.016	1.0e-4	10

Table I. A set of examples of searched configurations. Since PBT, PB2, BayesOpt, Dragonfly, ZOOpt can not select head, backbone, optimizer and batch size, they adopt the default configuration (the best configuration given by AME).