Appendices

A. Pathfinding Task Details

The Pathfinding graph is constrained to be a polytree (singly connected, directed acyclic graph) at each step of an episode, as outlined in Algorithm 1.

Algorithm 1 Pathfinding Episode Dynamics

Input: pattern size D, max graph size N.
Initialize $Graph = empty$.
Add a node with random pattern $\in (-1, +1)^D$.
$AddPattern = true$.
repeat
  Input: agent $Action$.
  $Reward = 0$.
  $Done = false$.
  if $AddPattern$ is $true$ **then**
    // Construction step.
    if $size(Graph) > 1$ and $Action == Target$ **then**
      $Reward = 1$.
    end if
    if $size(Graph) == N$ **then**
      $Done = true$.
    else
      Choose random node A from the Graph.
      Add node B with random pattern $\in (-1, +1)^D$.
      Link A and B in a random direction.
      $Observation = A, B, 0$.
      $AddPattern = false$.
    end if
  else
    // Quiz step.
    Choose random $Target \in (true, false)$.
    repeat
      Choose random nodes X and Y.
      $PathExists =$ path exists from X to Y.
    until $PathExists == Target$
    $Observation = X, Y, 1$.
    $AddPattern = true$.
  end if
  Output: $Reward, Observation, Done$
until $Done$ is $true$

The hand-coded baseline agent is configured with a depth parameter n. As new pattern pairs are revealed on graph-construction timesteps, this agent maintains a growing vector of all patterns seen, along with a growing matrix of directed path lengths from every observed pattern to every other. A path length of zero in this matrix indicates that no path exists from the first pattern to the second. On each quiz step, the agent looks up from the matrix the path length len for the ordered pair of patterns in the observation. If $0 < len \leq n$, the agent chooses the *yes* action. Otherwise, the agent chooses the *no* action.

B. Hyperparameter Tuning Procedures

B.1. DGD

In this work, all hyperparameters were tuned by a guided form of random search that we have named *Distributed Grid Descent* (DGD). It is designed to address the challenges posed by large numbers of hyperparameters (10-20), and the high variance among independent training runs for the same hyperparameter configuration that is often observed in Deep RL experiments. DGD tackles these challenges by steering the random selection of configurations to be tested towards a robust basin, a fixed point in configuration space for which modification of any individual (discrete) hyperparameter setting by one step higher or lower results in worse performance in expectation over repeated runs. In addition, DGD is designed to run on multiple processes on potentially many machines with no central point of control.

We define the following terms:

Tuning metric: A user-defined value calculated per training run for which higher is better, such as reward or success rate, or negative loss, etc.

Run result: A completed run's hyperparameter configuration and final tuning metric.

Run set: A single hyperparameter configuration, along with any available run results for that configuration.

Count(run set): The number of runs in a run set.

Metric(run set): The mean (or median) of the run metrics in a run set.

Neighborhood: A collection of run sets with configurations that differ by no more than one step in one setting from the configuration of a central run set in the neighborhood.

Count(neighborhood): The maximum $Count$ of all run sets in the neighborhood.

The operation of each DGD worker process is described by Algorithm 2.

Throughout the DGD search, the *best* current run set is determined through Bayesian inference based on each run set's $Metric$ and $Count$, to filter out high-variance run sets having high average scores but relatively few runs. After the best posterior performance remains stable for some number of runs, the DGD search is terminated, and the best run set's hyperparameter configuration is taken as the output of the search. To minimize the effects of local optima, the best run set can be chosen from a number of independent DGD searches.

Algorithm 2 Distributed Grid Descent

Input: Set of hyperparameters H, each having a discrete, ordered set of possible values.
Input: Maximum number of training steps N per run.
repeat
  Download any new run results.
  if no results so far **then**
    Choose a random configuration C from the grid defined by H.
  else
    Identify the run set S with the highest $Metric$.
    Initialize neighborhood B to contain only S.
    Expand B by adding all possible run sets whose configurations differ from that of S by one step in exactly one hyperparameter setting.
    Calculate a ceiling $M = Count(B) + 1$.
    Weight each run set x in B by $M - Count(x)$.
    Sample a random run set S' from B according to run set weights.
    Choose the configuration C from S'.
  end if
  Perform one training run of N steps using C.
  Calculate the run's $Metric$.
  Log the run result to shared storage.
until terminated by user.

B.2. Application of DGD

For the Pathfinding and BabyAI experiments, we ran five parallel DGD hyperparameter searches to convergence for each model, using the full number of training steps per run for the given experiment, and chose the best run set's hyperparameter configuration after convergence.

For WMG on Sokoban, we performed 60 DGD searches using training runs of 1.5M steps, with puzzle completion rate as the tuning metric. After those searches converged, we selected the best 25 configurations based on the training set results, and initiated 20 new training runs for each configuration. After training each agent for 10M environment interactions, we selected the single best hyperparameter configuration based on its performance on the training set. We then branched this configuration into 10 sets of runs with learning rate annealed every 100k steps using one of 10 separate values of gamma ranging from 0.6 to 0.98. After each agent was trained for a total of 20 million environment interactions, the annealing rate of 0.98 was found to perform the best on the training set. Finally, for this selected configuration's 20 independent agents, the models cached at 5M-step intervals were evaluated on the held-out test set of 1000 puzzles, producing the results shown in Figure 6 (right).

All experiments were performed on Linux virtual machines in the Microsoft Azure cloud. The virtual machines featured Intel 2.6GHz Xeon E5 2667 v3 processors with 8 virtual CPUs, and no GPUs.

C. Supplemental Tables

Table 2. Fixed settings and options used for all experiments, apart from the replicated BabyAI baselines in Table 15.

Settings and options	**Values**
Dropout	None
Learning rate schedule	Constant learning rate, except where noted
Non-linearities	ReLU, tanh
Parallel training workers	1
Optimizer	Adam (Kingma & Ba, 2014)
Parameter initialization, biases	0
Parameter initialization, non-bias weights	Kaiming uniform (He et al., 2015)
Reward shaping	None
Training algorithm	A3C (Mnih et al., 2016)
Weight decay regularization	None

Table 3. Hyperparameter values considered.

A3C t_{max}	1, 2, 3, 4, 6, 8, 10, 12, 16, 20, 24, 28, 32, 40, 48, 56, 64, 96, 120
Actor-critic hidden layer size	64, 90, 128, 180, 256, 360, 512, 720, 1024, 1440, 2048, 2880, 4096, 5760
Adam eps	1e-2, 1e-4, 1e-6, 1e-8, 1e-10, 1e-12
CNN channel size 1	12, 16, 20
CNN channel size 2	24, 32, 40
CNN channel size 3	64, 128, 192
Discount factor γ	0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.98, 0.99, 0.995, 0.998
Entropy term strength β	0.0, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2
Gradient clipping threshold	2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048
GRU observation embed size	128, 256, 512, 1024, 2048, 4096
GRU size	64, 96, 128, 192, 256, 384, 512, 768, 1024
Learning rate	4e-6, 6.3e-6, 1e-5, 1.6e-5, 2.5e-5, 4e-5, 6.3e-5, 1e-4, 1.6e-4, 2.5e-4, 4e-4
Learning rate annealing γ	0.60, 0.64, 0.68, 0.72, 0.76, 0.80, 0.84, 0.88, 0.93, 0.98
Reward on success (Sokoban)	2, 5, 10, 15, 20
Reward per step (Sokoban)	0, -0.01, -0.02
Reward scale factor	0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, 128
WMG attention head size	8, 12, 16, 24, 32, 48, 64, 90, 128, 180, 256, 360, 512
WMG attention heads	1, 2, 3, 4, 6, 8, 10, 12, 16, 20
WMG Memo size	32, 45, 64, 90, 128, 180, 256, 360, 512, 720, 1024, 1440, 2048
WMG Memos	1, 2, 3, 4, 6, 8, 10, 12, 16, 20
WMG hidden layer size	6, 8, 12, 16, 24, 32, 48, 64, 96, 128, 192, 256, 384, 512
WMG layers	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

For the Pathfinding domain, we verified in separate experiments that there is no difference in performance between setting the number of WMG Memos to 12 (the maximum episode length) or 16 (chosen by tuning). Since we introduce no penalty for model complexity, both DGD and random search would be expected to choose randomly between these possible values.

Table 4. Tuned hyperparameter settings for Pathfinding experiments of 20M steps.

	WMG	**nr-WMG**	**GRU**
Actor-critic hidden layer size	128	128	512
A3C t_{max}	16	16	16
Adam eps	1e-06	1e-08	1e-08
Discount factor γ	0.5	0.6	0.5
Entropy term strength β	0.01	0.005	0.02
Gradient clipping threshold	16.0	16.0	4.
GRU observation embedding size	-	-	256
GRU size	-	-	384
Learning rate	0.00016	0.00016	0.0001
Reward scale factor	2.0	1.0	0.5
WMG attention head size	12	16	-
WMG attention heads	6	6	-
WMG Memos	16	0	-
WMG Memo size	128	-	-
WMG hidden layer size	12	32	-
WMG layers	4	4	-

Table 5. Tuned hyperparameter settings for Pathfinding experiments of 1M steps.

	WMG	**nr-WMG**	**GRU**
Actor-critic hidden layer size	2880	256	5760
A3C t_{max}	16	16	16
Adam eps	1e-04	1e-12	1e-06
Discount factor γ	0.5	0.5	0.5
Entropy term strength β	0.005	0.05	0.1
Gradient clipping threshold	256.0	4.0	32.
GRU observation embedding size	-	-	1024
GRU size	-	-	256
Learning rate	0.00004	0.00016	0.0001
Reward scale factor	4.0	2.0	2.0
WMG attention head size	90	90	-
WMG attention heads	4	1	-
WMG Memos	6	0	-
WMG Memo size	256	-	-
WMG hidden layer size	8	16	-
WMG layers	3	5	-

Table 6. Tuned hyperparameter settings for BabyAI Level 1 - GoToObj.

	WMG factored	**nr-WMG factored**	**GRU factored**	**WMG flat**	**GRU flat**	**CNN+GRU native 7x7x3**
Actor-critic hidden layer size	2048	4096	4096	4096	2048	512
A3C t_{max}	1	1	6	16	4	6
Adam eps	0.0001	1e-08	1e-08	1e-10	0.0001	1e-10
CNN hidden channel size 1	-	-	-	-	-	16
CNN hidden channel size 2	-	-	-	-	-	40
CNN hidden channel size 3	-	-	-	-	-	192
Discount factor γ	0.98	0.9	0.7	0.6	0.9	0.8
Entropy term strength β	0.002	0.05	0.01	0.005	0.02	0.02
Gradient clipping threshold	256.0	1024.0	512.0	512.0	128.0	128.0
GRU observation embed size	-	-	1024	-	512	512
GRU size	-	-	96	-	512	96
Learning rate	0.0001	4e-05	0.0004	0.0001	0.0001	0.0004
Reward scale factor	4.0	32.0	32.0	8.0	32.0	8.0
WMG attention head size	24	16	-	16	-	-
WMG attention heads	4	10	-	12	-	-
WMG Memos	1	0	-	1	-	-
WMG Memo size	64	-	-	256	-	-
WMG hidden layer size	64	64	-	32	-	-
WMG layers	4	4	-	1	-	-

Table 7. Tuned hyperparameter settings for BabyAI Level 2 - GoToRedBallGrey.

	WMG factored	**nr-WMG factored**	**GRU factored**	**WMG flat**	**GRU flat**	**CNN+GRU native 7x7x3**
Actor-critic hidden layer size	4096	2048	4096	4096	4096	64
A3C t_{max}	8	6	16	1	1	1
Adam eps	1e-06	1e-08	1e-10	1e-10	1e-06	0.0001
CNN hidden channel size 1	-	-	-	-	-	12
CNN hidden channel size 2	-	-	-	-	-	24
CNN hidden channel size 3	-	-	-	-	-	192
Discount factor γ	0.8	0.9	0.8	0.9	0.9	0.95
Entropy term strength β	0.01	0.02	0.01	0.005	0.005	0.02
Gradient clipping threshold	1024.0	512.0	1024.0	128.0	64.0	64.0
GRU observation embed size	-	-	4096	-	2048	256
GRU size	-	-	96	-	512	64
Learning rate	0.0001	0.00025	0.0001	2.5e-05	2.5e-05	0.0004
Reward scale factor	8.0	4.0	4.0	4.0	4.0	2.0
WMG attention head size	64	48	-	64	-	-
WMG attention heads	4	1	-	3	-	-
WMG Memos	1	0	-	8	-	-
WMG Memo size	32	-	-	64	-	-
WMG hidden layer size	16	24	-	384	-	-
WMG layers	3	3	-	1	-	-

Table 8. Tuned hyperparameter settings for BabyAI Level 3 - GoToRedBall.

	WMG factored	**nr-WMG factored**	**GRU factored**	**WMG flat**	**GRU flat**	**CNN+GRU native 7x7x3**
Actor-critic hidden layer size	4096	2048	4096	4096	4096	4096
A3C t_{max}	1	2	3	1	2	3
Adam eps	1e-12	0.0001	1e-06	0.0001	1e-06	0.01
CNN hidden channel size 1	-	-	-	-	-	12
CNN hidden channel size 2	-	-	-	-	-	40
CNN hidden channel size 3	-	-	-	-	-	192
Discount factor γ	0.95	0.9	0.9	0.9	0.9	0.9
Entropy term strength β	0.1	0.05	0.1	0.05	0.02	0.05
Gradient clipping threshold	128.0	128.0	128.0	128.0	32.0	32.0
GRU observation embed size	-	-	2048	-	4096	256
GRU size	-	-	192	-	512	64
Learning rate	2.5e-05	6.3e-05	6.3e-05	2.5e-05	2.5e-05	0.0004
Reward scale factor	8.0	4.0	8.0	8.0	4.0	4.0
WMG attention head size	128	32	-	24	-	-
WMG attention heads	2	8	-	12	-	-
WMG Memos	2	0	-	16	-	-
WMG Memo size	128	-	-	256	-	-
WMG hidden layer size	64	32	-	128	-	-
WMG layers	4	4	-	1	-	-

Table 9. Tuned hyperparameter settings for BabyAI Level 4 - GoToLocal.

	WMG factored	**nr-WMG factored**	**GRU factored**	**WMG flat**	**GRU flat**
Actor-critic hidden layer size	2048	2048	1024	512	4096
A3C t_{max}	6	3	3	6	4
Adam eps	1e-12	0.01	1e-06	1e-08	1e-12
Discount factor γ	0.5	0.6	0.95	0.5	0.9
Entropy term strength β	0.1	0.1	0.1	0.02	0.02
Gradient clipping threshold	512.0	512.0	256.0	256.0	512.0
GRU observation embed size	-	-	1024	-	512
GRU size	-	-	128	-	96
Learning rate	6.3e-05	0.0001	4e-05	2.5e-05	4e-05
Reward scale factor	32.0	16.0	8.0	16.0	2.0
WMG attention head size	128	64	-	24	-
WMG attention heads	2	4	-	16	-
WMG Memos	8	0	-	16	-
WMG Memo size	32	-	-	64	-
WMG hidden layer size	32	48	-	16	-
WMG layers	4	3	-	2	-

Table 10. Tuned hyperparameter settings for BabyAI Level 5 - PickupLoc.

	WMG factored	**nr-WMG factored**
Actor-critic hidden layer size	512	2048
A3C t_{max}	12	12
Adam eps	1e-10	1e-10
Discount factor γ	0.7	0.8
Entropy term strength β	0.02	0.05
Gradient clipping threshold	512.0	512.0
Learning rate	0.0001	6.3e-05
Reward scale factor	8.0	8.0
WMG attention head size	24	48
WMG attention heads	10	6
WMG Memos	8	0
WMG Memo size	32	-
WMG hidden layer size	128	96
WMG layers	2	2

Table 11. Tuned hyperparameter values on Sokoban. The resulting model contained 4,508,182 trainable parameters.

Reward per step	0
Reward on success	2
Actor-critic hidden layer size	2880
A3C t_{max}	4
Adam eps	1e-10
Discount factor γ	0.995
Entropy term strength β	0.02
Gradient clipping threshold	512.0
Learning rate	1.6e-5
Learning rate annealing γ	0.98
Reward scale factor	4
WMG attention head size	32
WMG attention heads	8
WMG Memos	1
WMG Memo size	2048
WMG hidden layer size	8
WMG layers	10

Table 12. Additional details for Pathfinding experiments of 20M steps.

Models & algorithms	Final performance	Trainable parameters	Training speed
Depth-(n-1) baseline	100.0% of reward		
Depth-3 baseline	99.7% of reward		
Depth-2 baseline	97.6% of reward		
Depth-1 baseline	86.9% of reward		
nr-WMG, full-history	99.6% of reward	204,963	96 steps/sec
WMG	99.6% of reward	132,507	91 steps/sec
GRU	94.7% of reward	1,139,459	291 steps/sec

Table 13. Number of trainable parameters, in thousands, for the BabyAI models in Table 1.

BabyAI level	WMG factored	nr-WMG factored	GRU factored	WMG flat	GRU flat	CNN+GRU native 7x7x3
1 - GoToObj	636	1,864	1,572	2,053	4,170	393
2 - GoToRedBallGrey	2,997	258	3,723	2,116	10,075	140
3 - GoToRedBall	3,418	2,217	3,749	3,229	15,126	709
4 - GoToLocal	2,235	1,960	1,137	2,022	1,479	—
5 - PickupLoc	879	2,007	—	—	—	—

Table 14. Training steps per second on a fixed machine, for the BabyAI models in Table 1.

BabyAI level	WMG factored	nr-WMG factored	GRU factored	WMG flat	GRU flat	CNN+GRU native 7x7x3
1 - GoToObj	38	28	146	111	86	149
2 - GoToRedBallGrey	58	113	147	35	18	88
3 - GoToRedBall	18	32	78	25	20	87
4 - GoToLocal	44	48	132	54	134	—
5 - PickupLoc	81	84	—	—	—	—

Table 15. BabyAI baseline agent sample efficiencies, defined as the amount of training (in either episodes or environment interaction steps) required for the agent to solve 99% of random episodes within 64 steps. The published results are the means of the min & max RL sample efficiencies reported in Table 3 of Chevalier-Boisvert et al. (2018). We obtained the replicated results, which are the medians over 10 training runs, using the code and default hyperparameter settings from the open source release of the BabyAI baseline agent. We report these sample efficiencies in terms of both episodes and environment interactions. All numbers are in thousands.

BabyAI level	Instruction template	Published (*episodes*)	Replicated (*episodes*)	Replicated (env interactions)
1 - GoToObj	GO TO ⟨color⟩ ⟨object⟩	—	*19*	333
2 - GoToRedBallGrey	GO TO RED BALL	*17*	*16*	282
3 - GoToRedBall	GO TO RED BALL	*297*	*283*	3,674
4 - GoToLocal	GO TO ⟨color⟩ ⟨object⟩	*1008*	*1,064*	16,422
5 - PickupLoc	PICK UP ⟨color⟩ ⟨object⟩ ⟨location⟩	*1,545*	*1,557*	25,574