Continual and Multi-Task Architecture Search

Ramkanth Pasunuru

www.rama-kanth.com

Mohit Bansal

www.cs.unc.edu/~mbansal/

UNC NLP

Introduction

In this work, we extend architecture search approach to two important paradigms of transfer learning.

Introduction

In this work, we extend architecture search approach to two important paradigms of transfer learning.

Continual Learning

Model parameters evolve and adapt when trained sequentially on a new task

Introduction

In this work, we extend architecture search approach to two important paradigms of transfer learning.

Continual Learning	**Model parameters evolve and adapt when trained sequentially on a new task**
Multi-Task Learning	**Given multiple tasks in parallel, learns a generalizable cell structure**

Continual Learning

Definition: Continual learning (CL) is the ability to learn continually from a stream of data, building on what was learnt previously, while being able to *reapply*, *adapt* and *generalize* it to new situations.

[Donahue et al., 2014; Zeneke et al., 2017; Yoon et al., 2018]

Continual Learning

Definition: Continual learning (CL) is the ability to learn continually from a stream of data, building on what was learnt previously, while being able to *reapply*, *adapt* and *generalize* it to new situations.

Key Challenges:

Transfer and Adapt

[Donahue et al., 2014; Zeneke et al., 2017; Yoon et al., 2018]

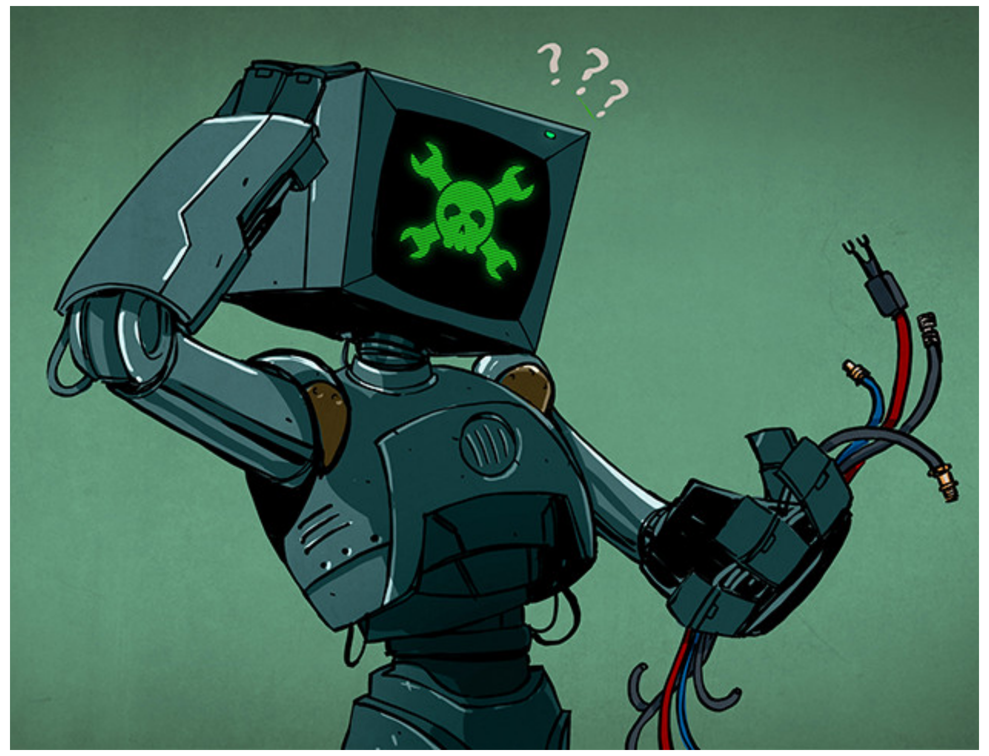
Continual Learning

Definition: Continual learning (CL) is the ability to learn continually from a stream of data, building on what was learnt previously, while being able to *reapply*, *adapt* and *generalize* it to new situations.

Key Challenges:

Transfer and Adapt

Catastrophic Forgetting



[Donahue et al., 2014; Zeneke et al., 2017; Yoon et al., 2018]

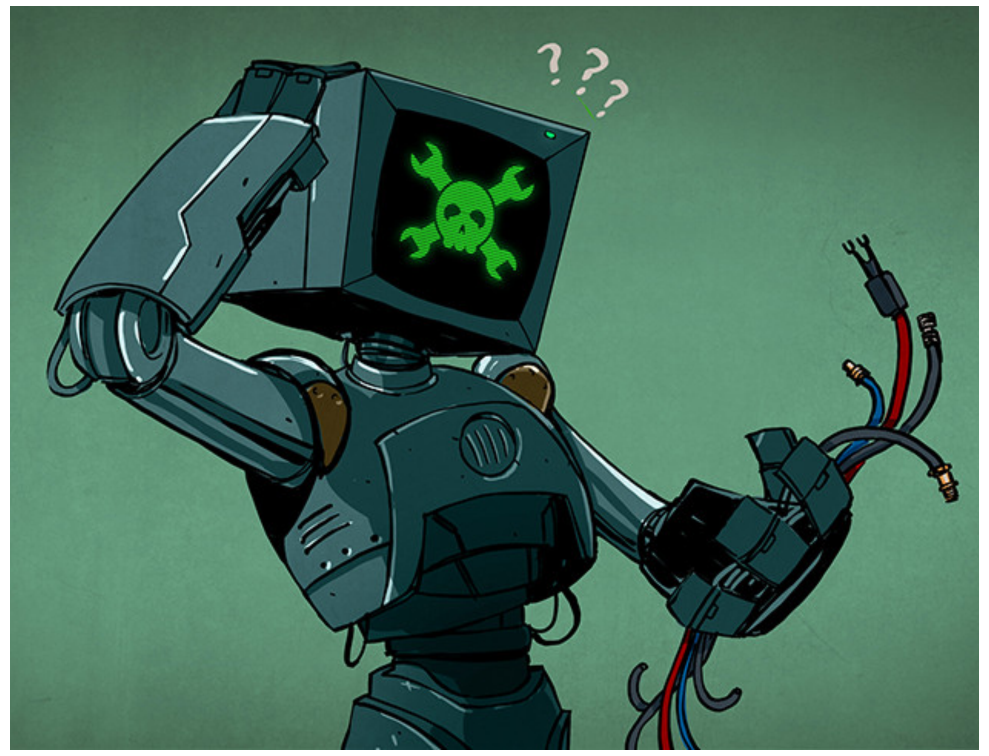
Continual Learning

Definition: Continual learning (CL) is the ability to learn continually from a stream of data, building on what was learnt previously, while being able to *reapply*, *adapt* and *generalize* it to new situations.

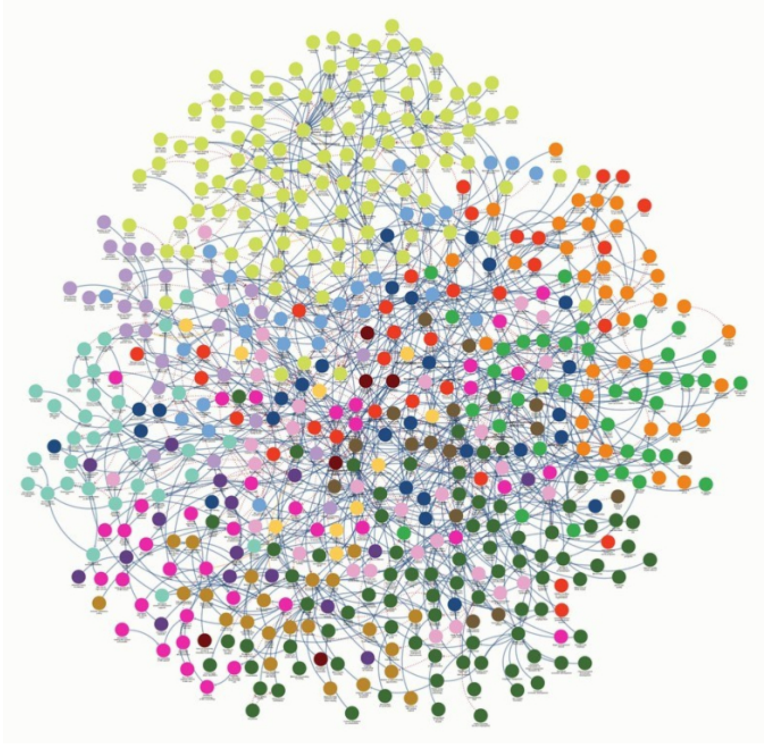
Key Challenges:

Transfer and Adapt

Catastrophic Forgetting



Bounded System Size



[Donahue et al., 2014; Zeneke et al., 2017; Yoon et al., 2018]

Previous Work

Regularization to penalize functional or shared parameters' change
[Razavian et al., 2014; Li and Hoiem, 2017; Hinton et al., 2015; Jung et al., 2016; Kirk- patrick et al., 2017; Donahue et al., 2014; Yosinski et al., 2014]

Copying the previous task and augmenting with new task's features
[Rusu et al., 2016]

Intelligent synapses to accumulate task-related information
[Zeneke et al., 2017]

Dynamically expandable network based on incoming new data
[Yoon et al., 2018]

Idea!

Leverage Neural Architecture Search!

Continual Architecture Search (CAS): continually evolve the model parameters during the sequential training of several tasks by leveraging neural architecture search.

[Donahue et al., 2014; Zeneke et al., 2017; Yoon et al., 2018]

NAS

Neural Architecture Search (NAS): has been recently introduced for automatic learning of the model structure for the given dataset/task.

- Shown good improvements on image classification and language modeling
- Computationally feasible NAS approaches:
  - Tree-structured search space
  - ϵ-greedy exploration

[Zoph & Le, 2017; Baker et al., 2017; Negrinho & Gordon, 2017]

ENAS

Neural Architecture Search (NAS): has been recently introduced for automatic learning of the model structure for the given dataset/task.

- Shown good improvements on image classification and language modeling
- Computationally feasible NAS approaches:
  - Tree-structured search space
  - ϵ-greedy exploration

Efficient Neural Architecture Search (ENAS): A weight-sharing strategy among search parameters **[Pham et al., 2018]**

[Zoph & Le, 2017; Baker et al., 2017; Negrinho & Gordon, 2017]

ENAS

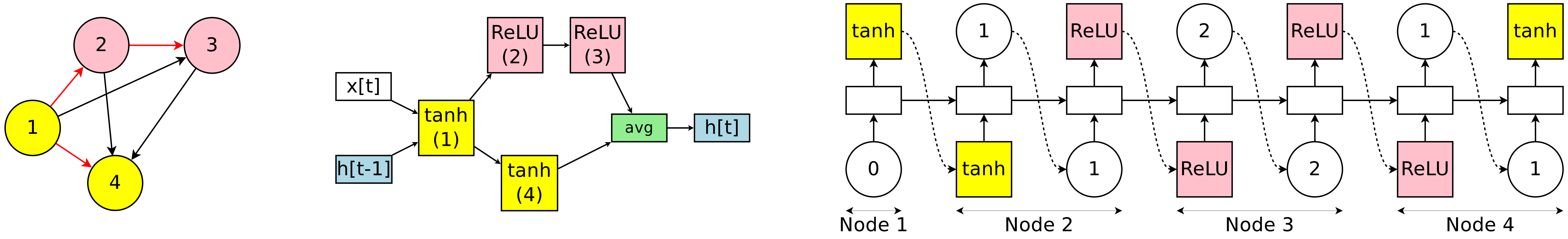


Figure: An example of a recurrent cell in our search space with 4 computational nodes. *Left:* The computational DAG that corresponds to the recurrent cell. The red edges represent the flow of information in the graph. *Middle:* The recurrent cell. *Right:* The outputs of the controller RNN that result in the cell in the middle and the DAG on the left. Note that nodes 3 and 4 are never sampled by the RNN, so their results are averaged and are treated as the cell's output.

[Pham et al., 2018]

ENAS

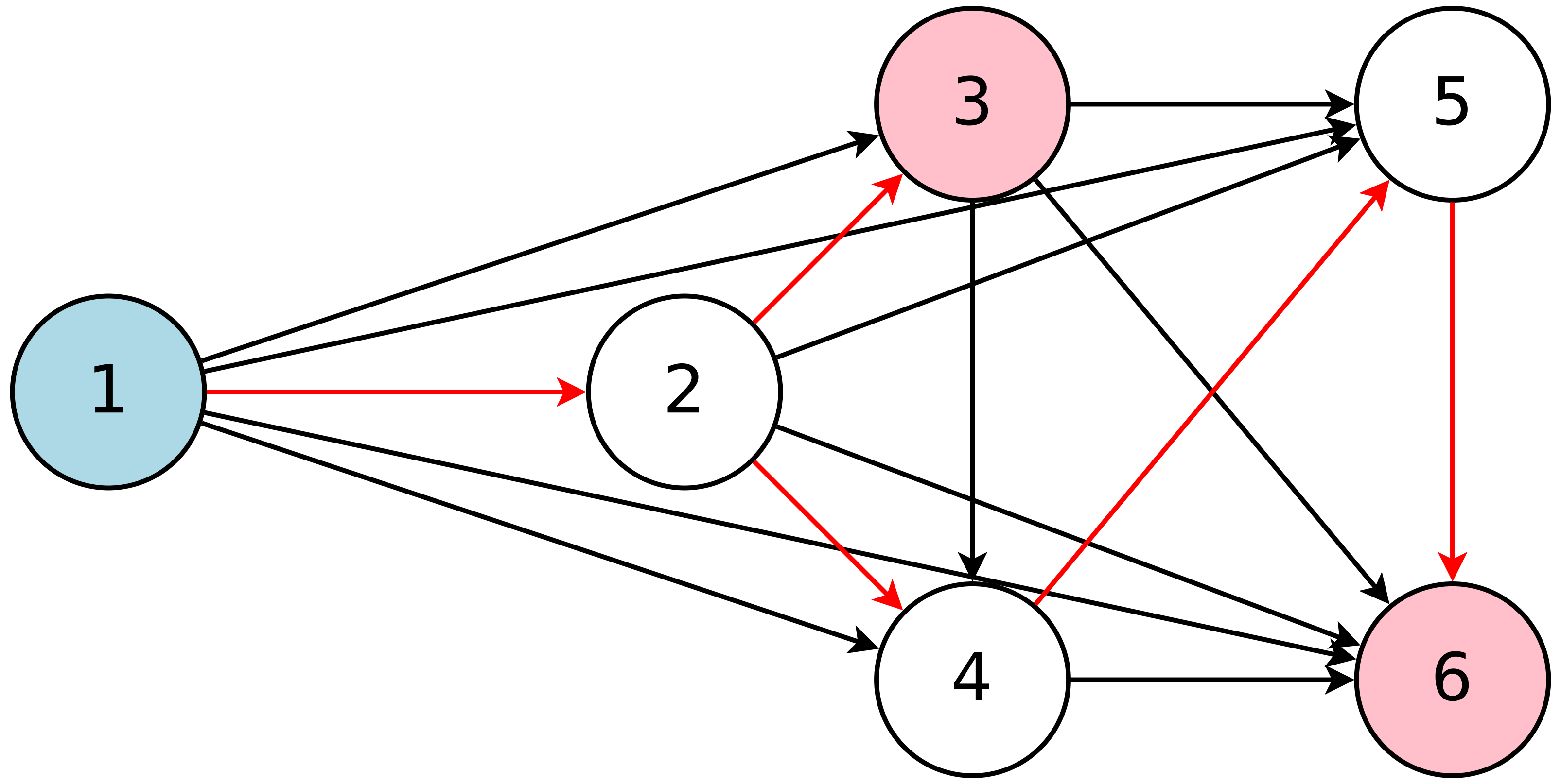
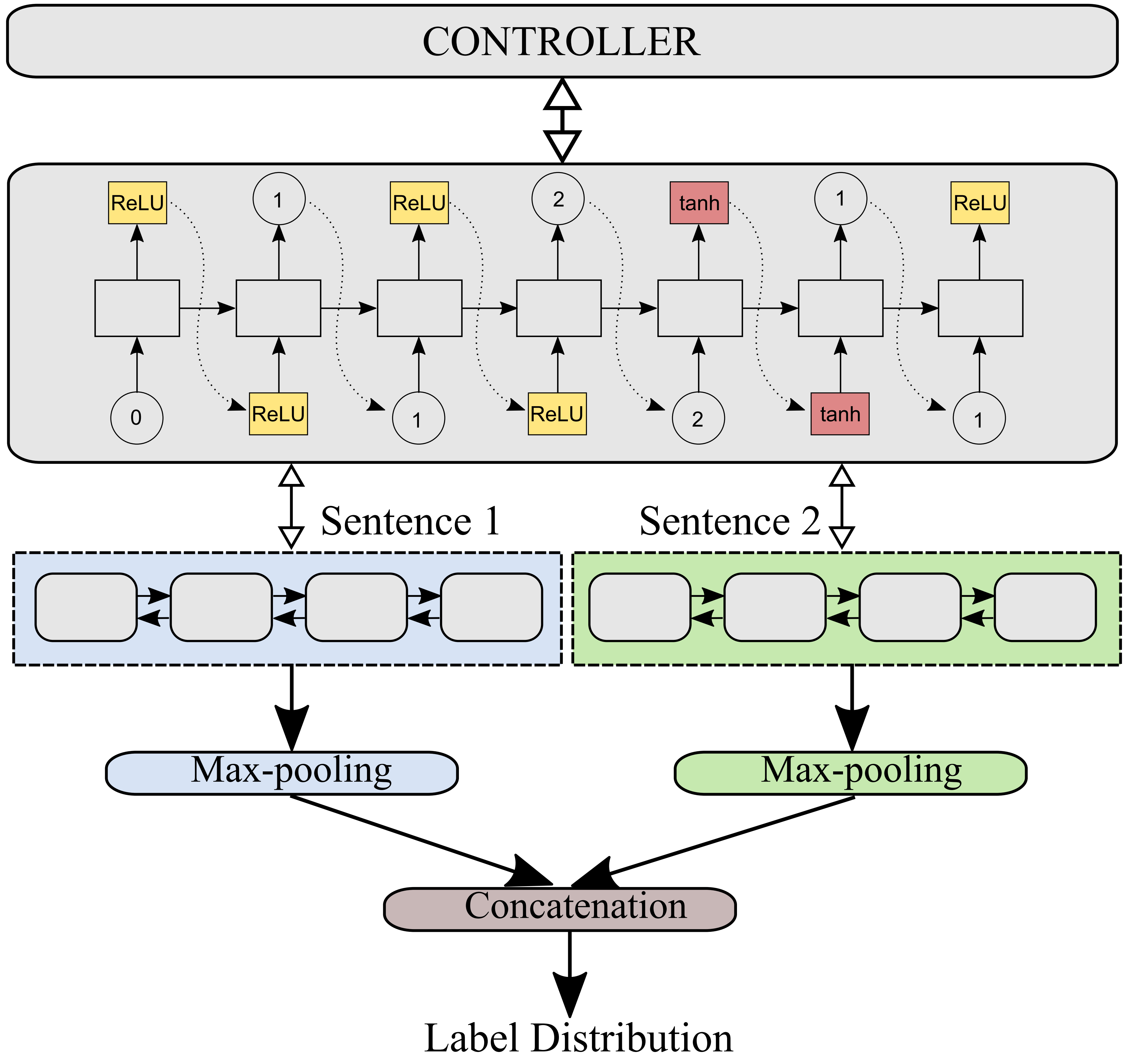


Figure: The graph represents the entire search space while the red arrows define a model in the search space, which is decided by a controller. Here, node 1 is the input to the model whereas nodes 3 and 6 are the model's outputs.

[Pham et al., 2018]

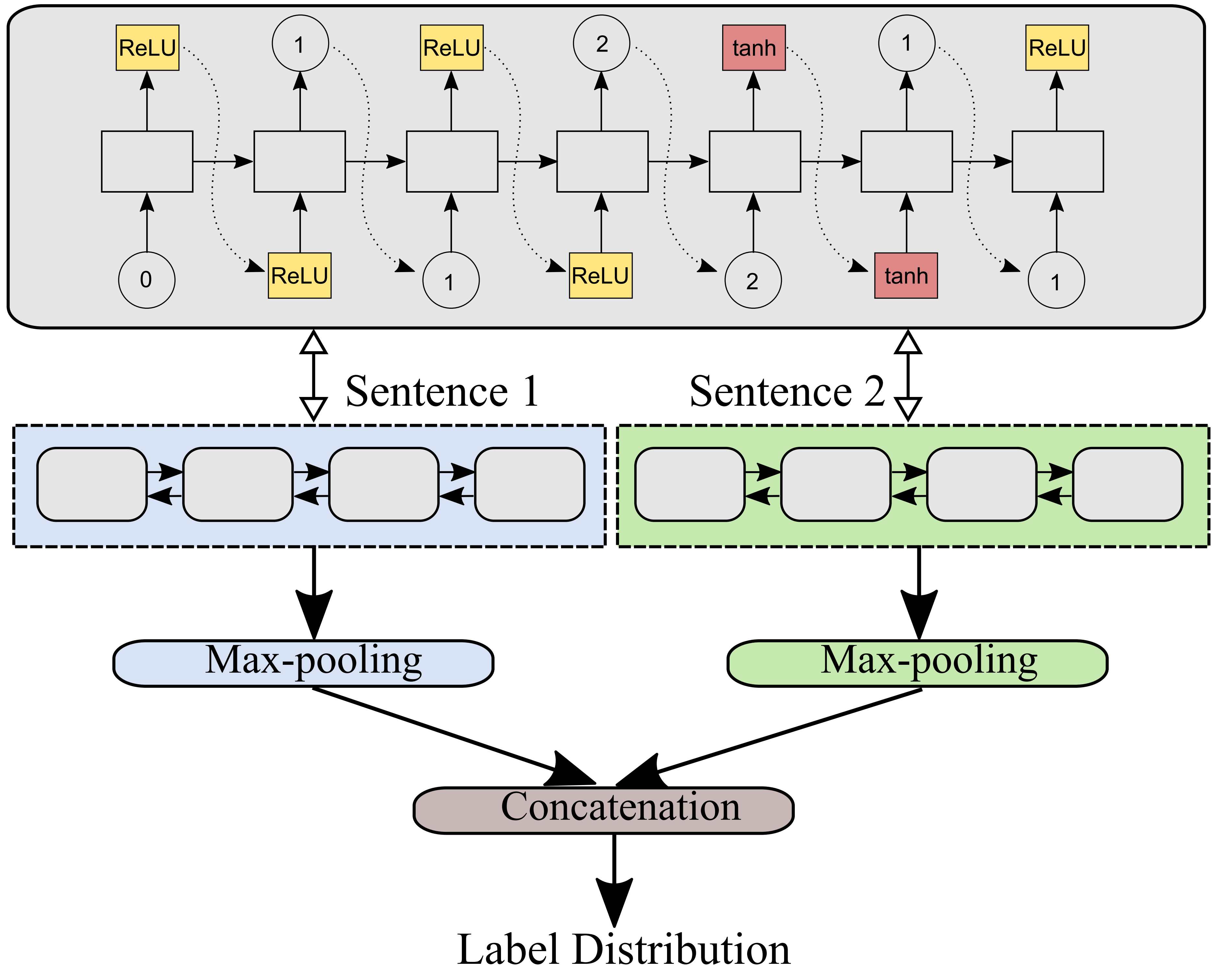
ENAS for Text Classification



Stage1:

- Controller samples a cell structure and use the task's performance as feedback
- Controller learns optimal cell structure

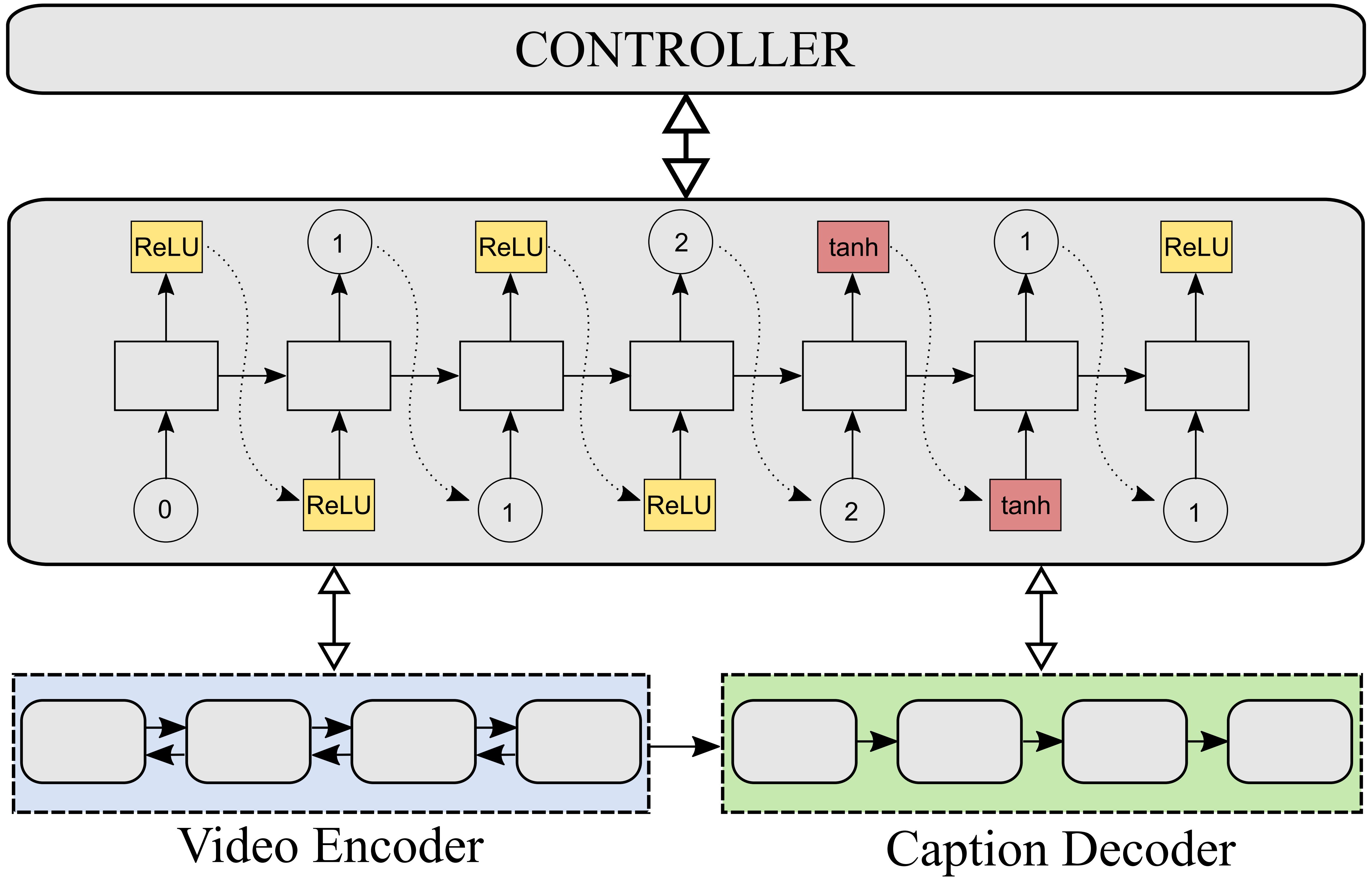
ENAS for Text Classification



Stage2:

- Retrain the model using the learned optimal cell structure in stage-1

ENAS for Sequence Generation



Continual Architecture Search (CAS)

Task1 (dataset d_1)

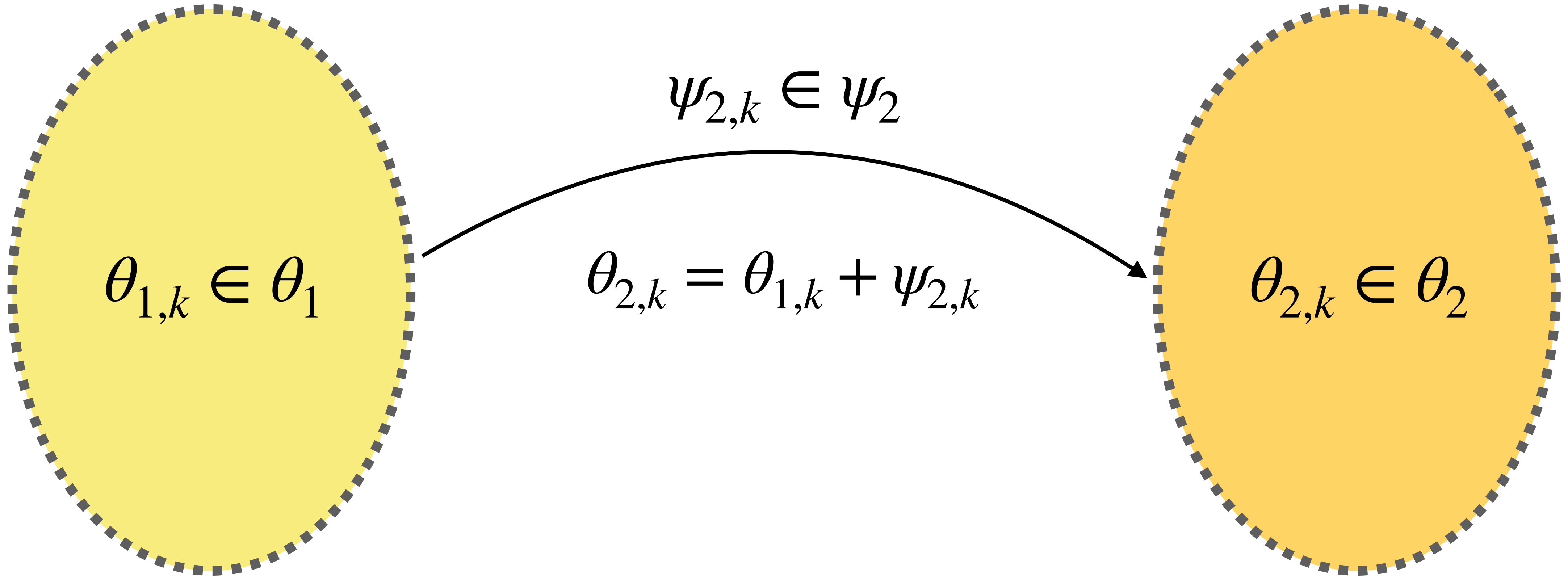
$\theta_{1,k} \in \theta_1$

Task2 (dataset d_2)

$\theta_{2,k} \in \theta_2$

Continual Architecture Search (CAS)

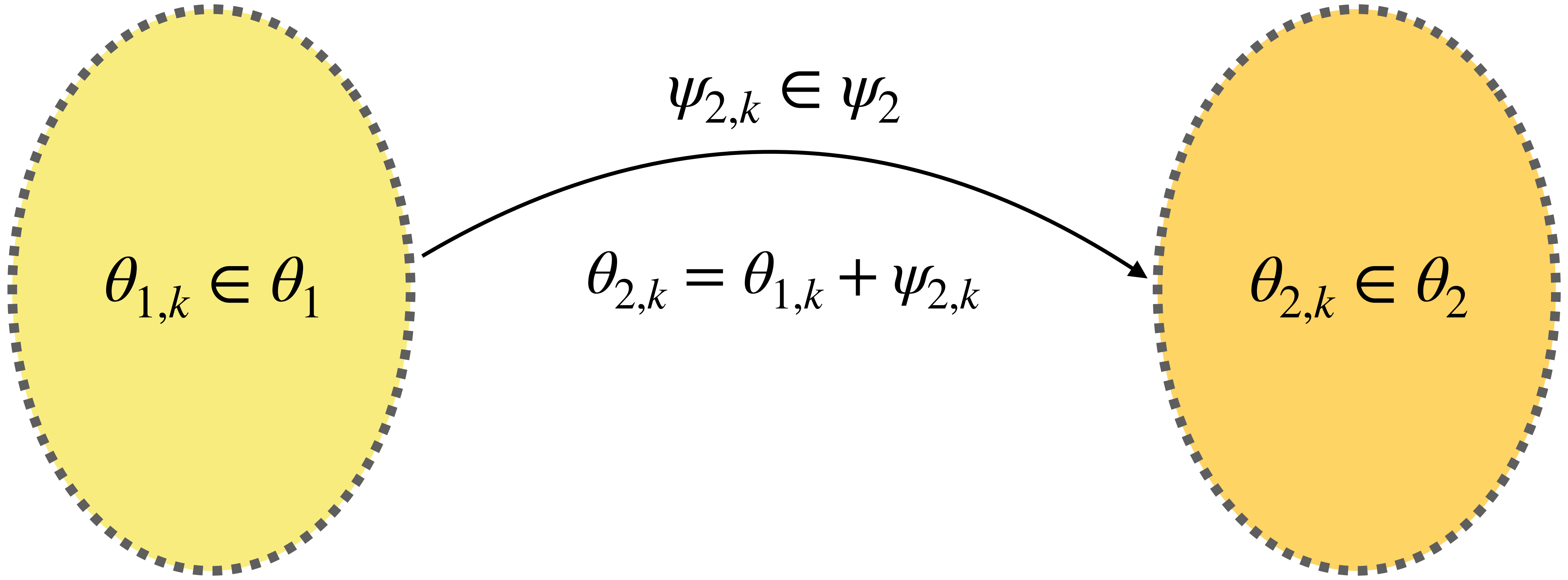
Task1 (dataset d_1) **Task2** (dataset d_2)



Continual Architecture Search (CAS)

Task1 (dataset d_1)

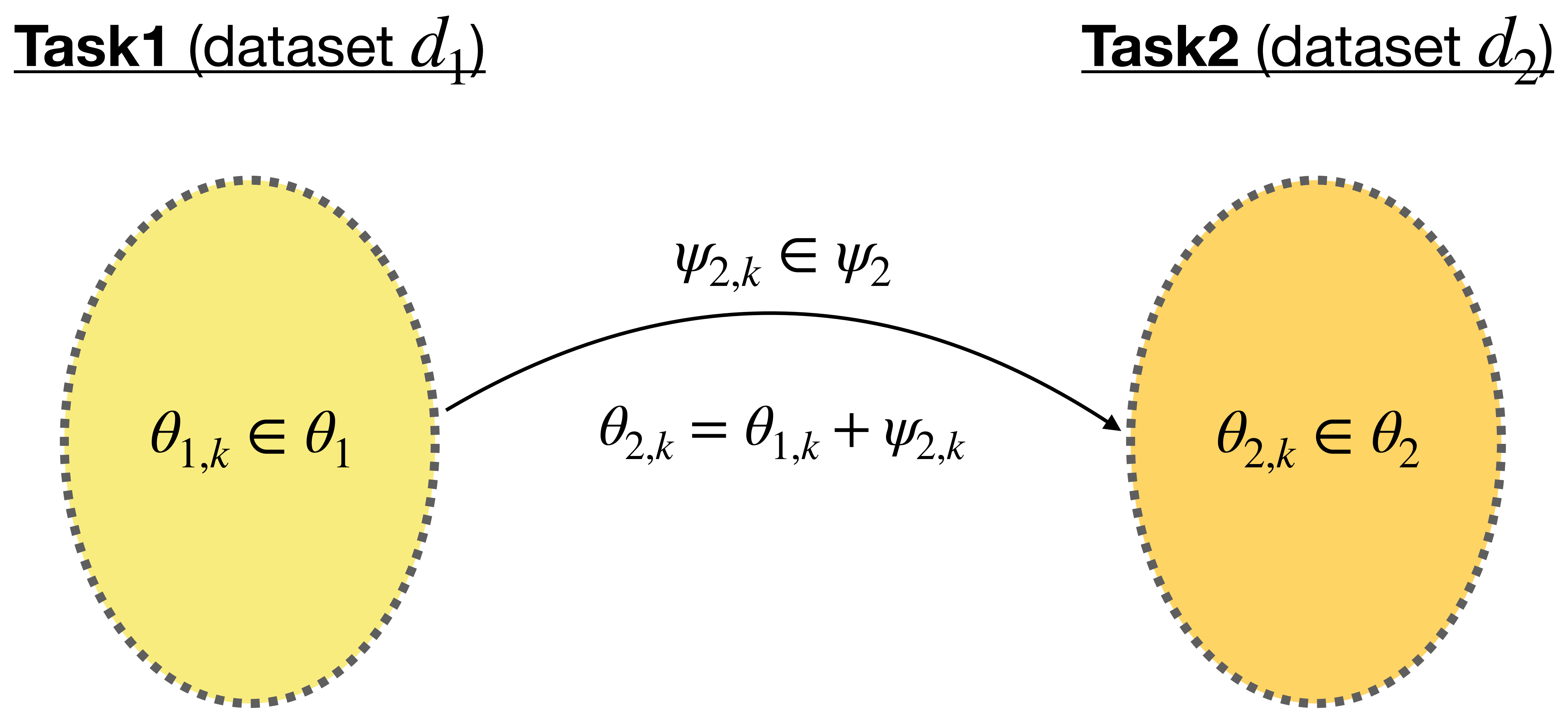
Task2 (dataset d_2)



θ_1 are block-sparse in nature

[Scardapane et al., 2017]

Continual Architecture Search (CAS)

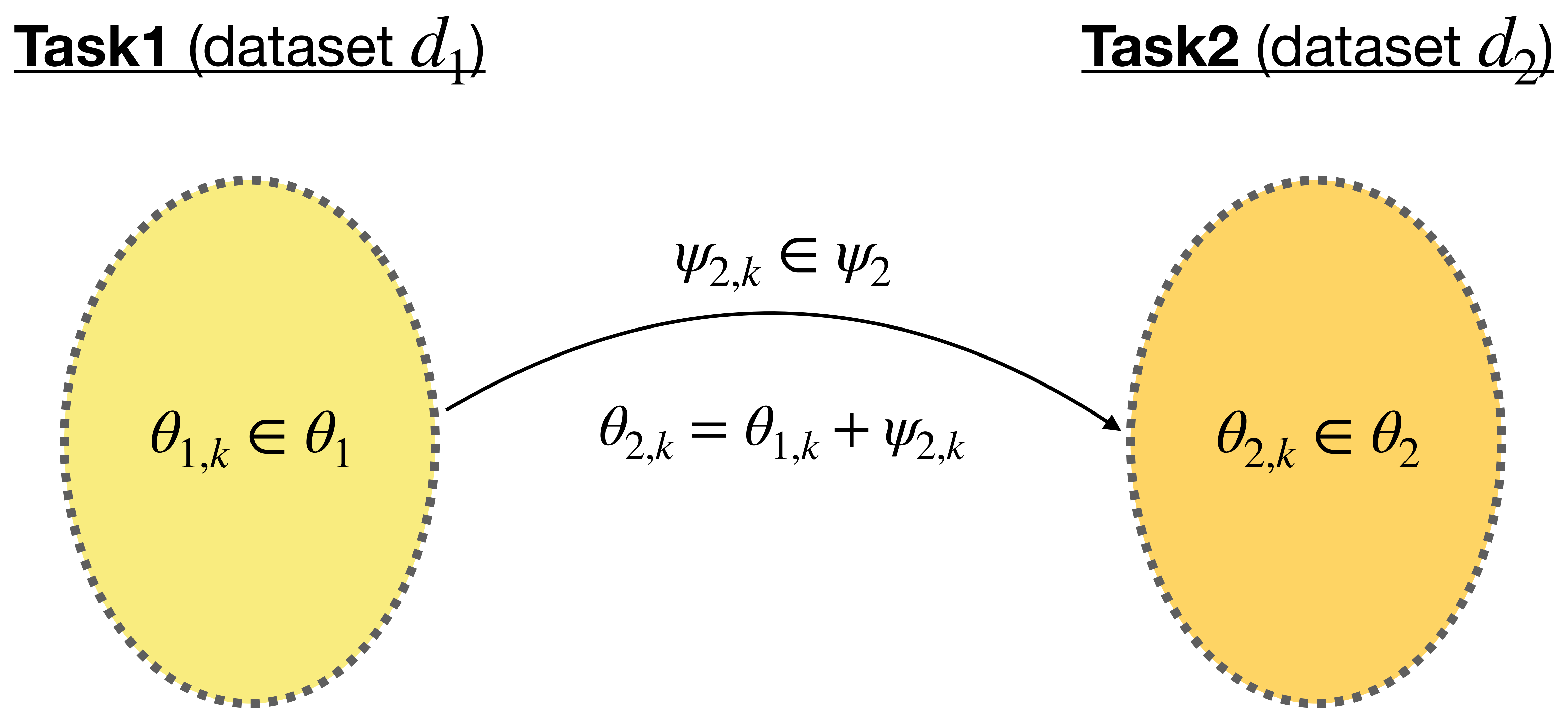


θ_1 are block-sparse in nature

ψ_2 is orthogonal to θ_1

[Bousmalis et al., 2016; Scardapane et al., 2017]

Continual Architecture Search (CAS)



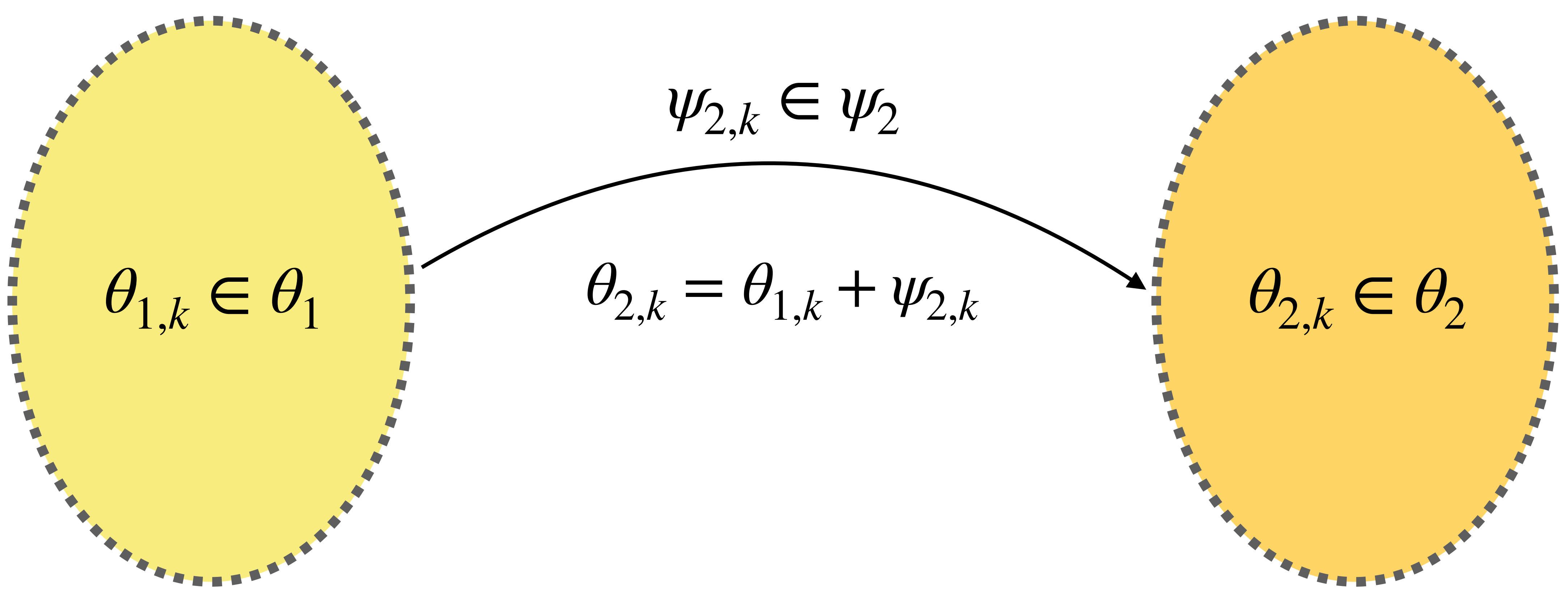
Condition 1: *When training the model on dataset* d_1, *we constrain the model parameters* $\theta_{1,k} \in R^{m\times n}$ *to be sparse, specifically, to be block sparse, i.e., minimize*

$$\sum_{i=1}^{m} |(||\theta_{1,k}[i,:]||_2)|_1$$

[Scardapane et al., 2017]

Continual Architecture Search (CAS)

Task1 (dataset d_1) **Task2** (dataset d_2)

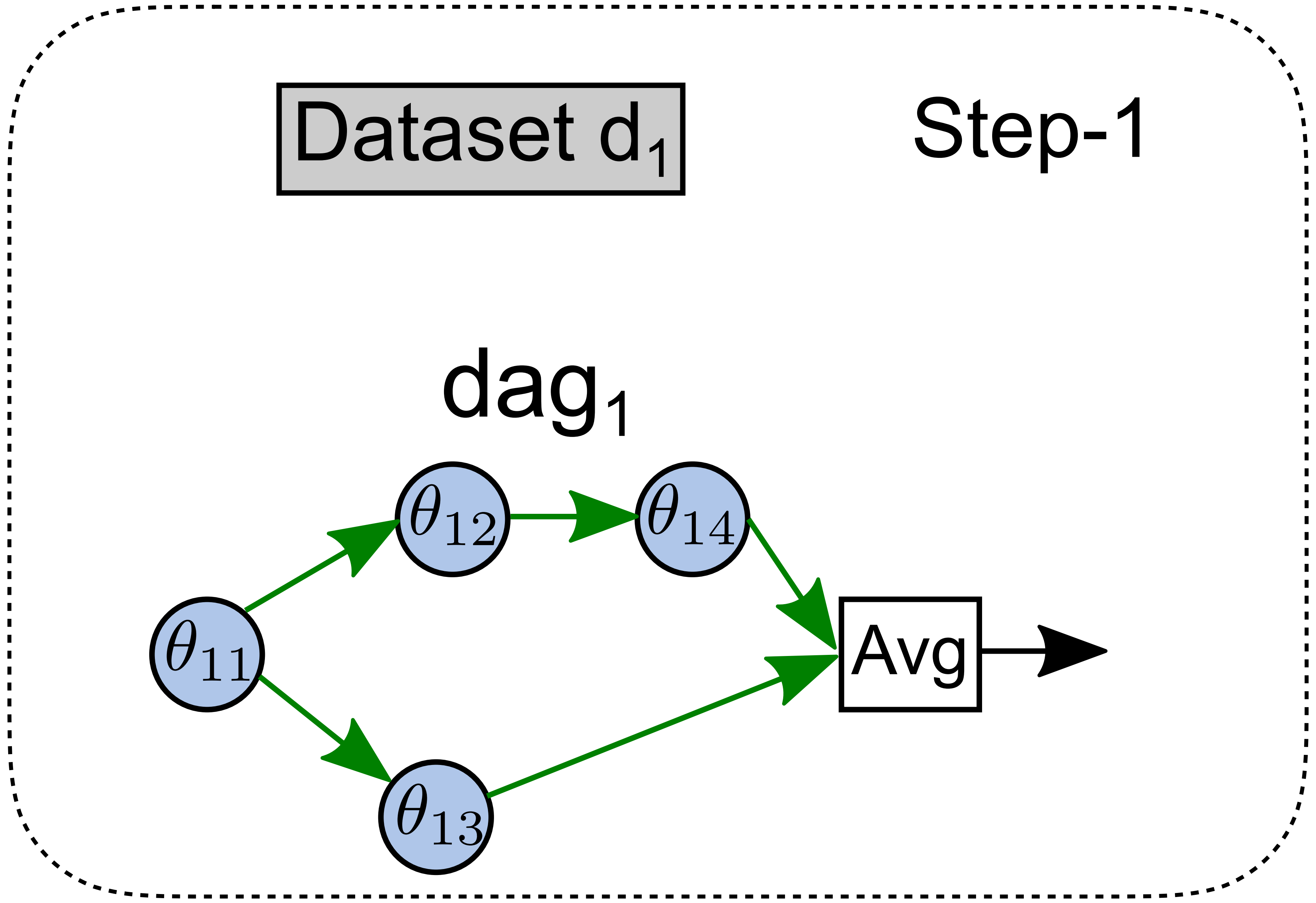


Condition 2: *When training the model on dataset d_2, we start from $\theta_{1,k}$, keep it constant, and update $\psi_{2,k}$ such that:*

1. $\psi_{2,k}$ *is block sparse, i.e.,* $\sum_{i=1}^{m} |(||\psi_{2,k}[i,:]||_2)|_1$.
2. $\theta_{1,k}$ *and* $\psi_{2,k}$ *are orthogonal.*

[Bousmalis et al., 2016]

CAS Step-1



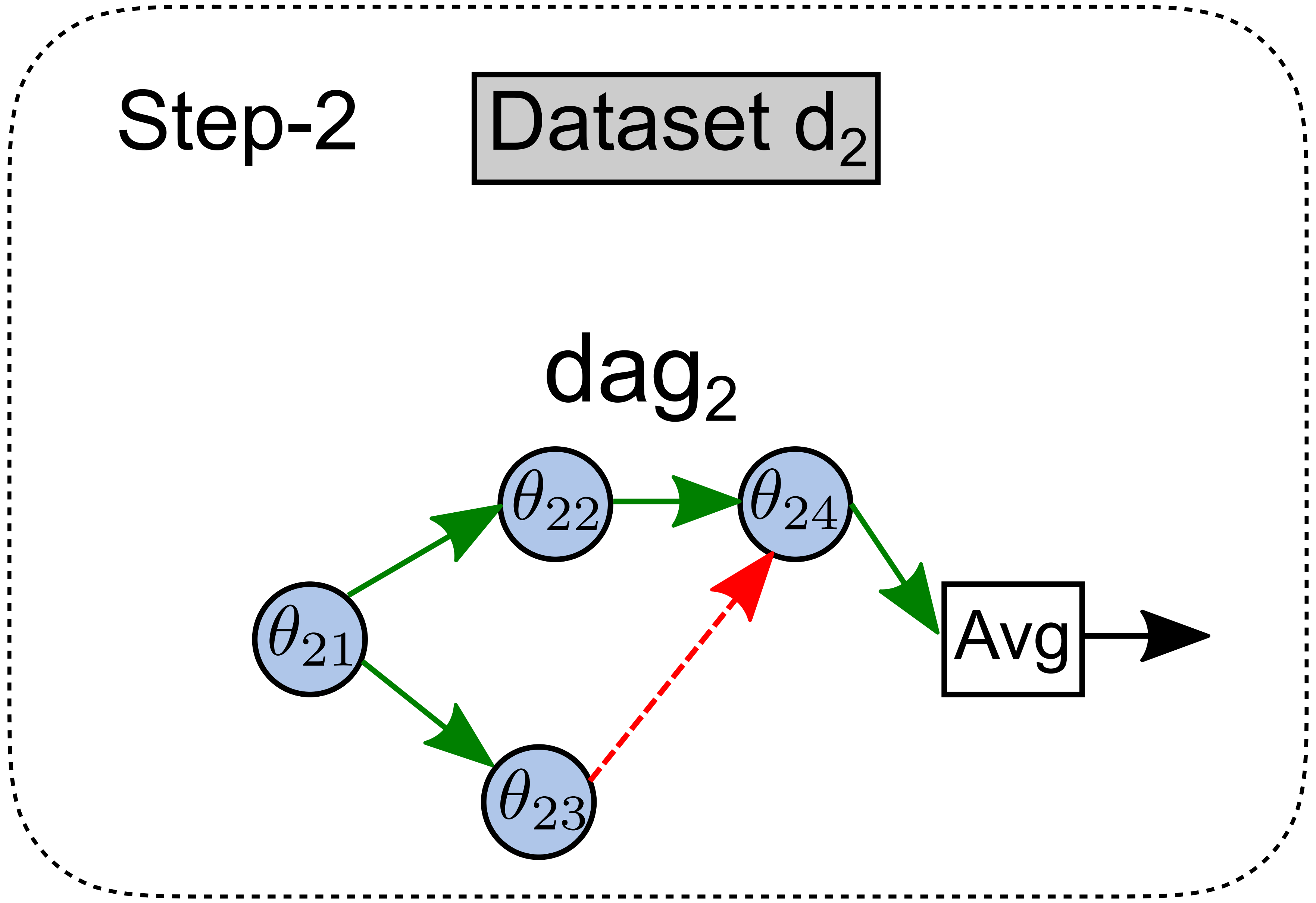
Train on Dataset d_1

Apply Condition 1

Learn parameters θ_1 and cell structure dag_1

R. Pasunuru & M. Bansal

CAS Step-2



Train on Dataset d_2 and initialize the parameters with θ_1

Apply Condition 2

Learn parameters θ_2 and cell structure dag_2

CAS Evaluation

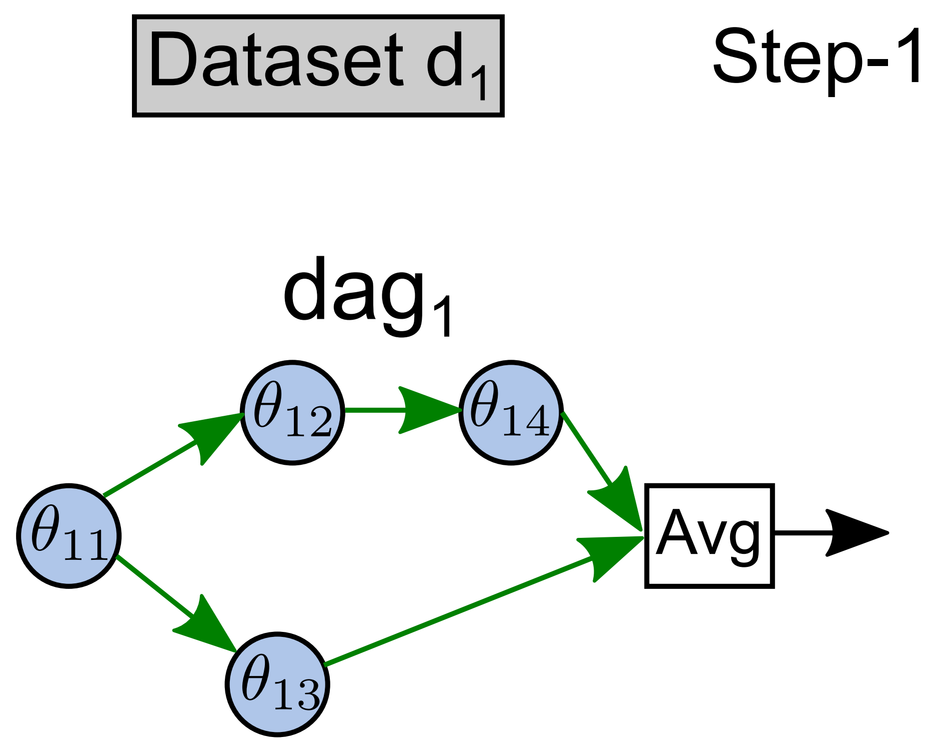


Figure: Continual architecture search (CAS) approach: green, solid edges (weight parameters) are shared, newly-learned edges are represented with red, dashed edges.

CAS Evaluation

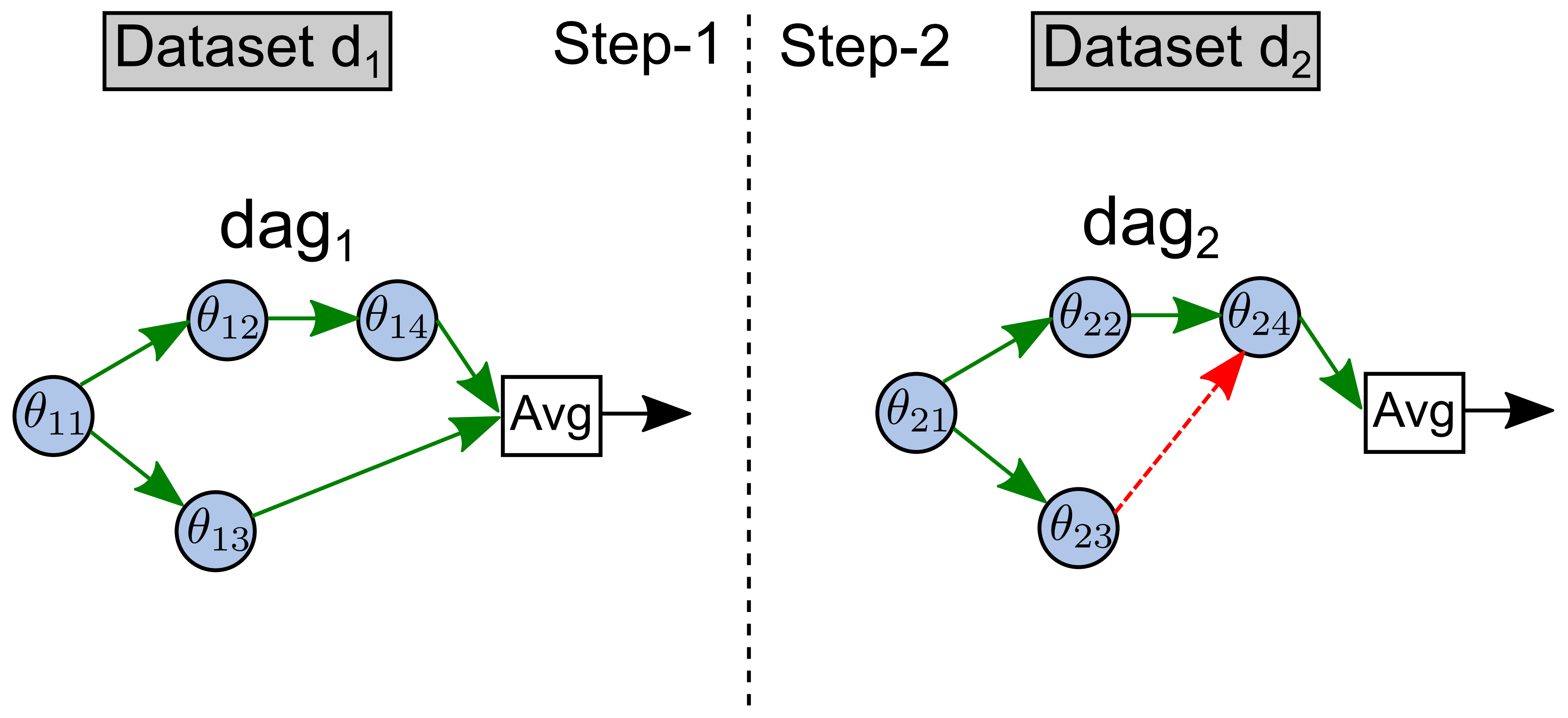


Figure: Continual architecture search (CAS) approach: green, solid edges (weight parameters) are shared, newly-learned edges are represented with red, dashed edges.

CAS Evaluation

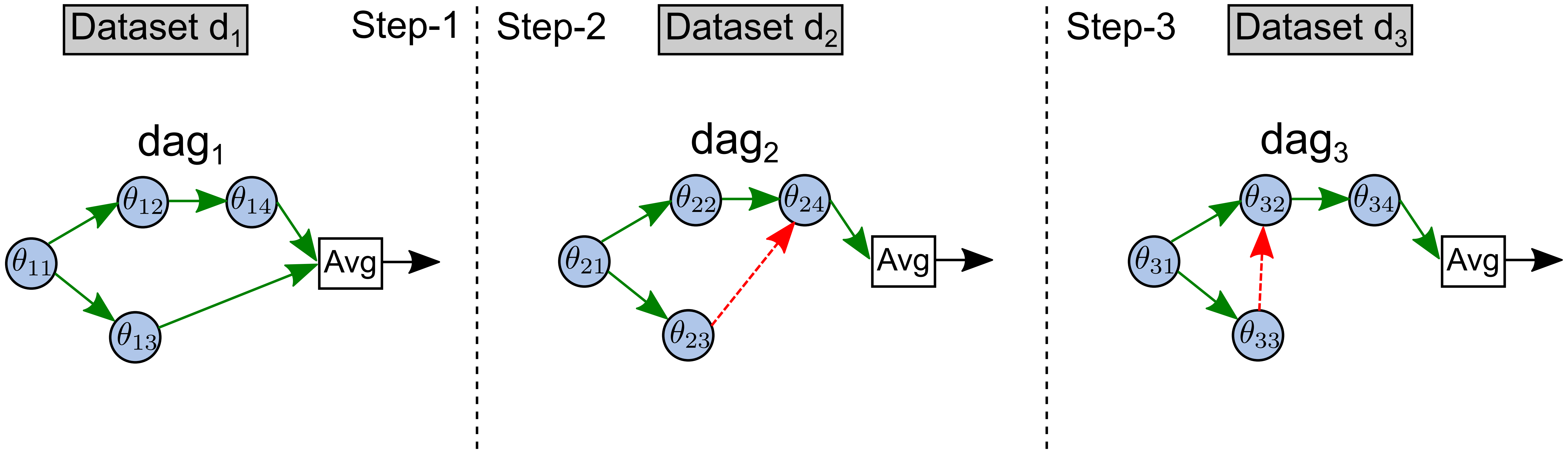


Figure: Continual architecture search (CAS) approach: green, solid edges (weight parameters) are shared, newly-learned edges are represented with red, dashed edges.

CAS Evaluation

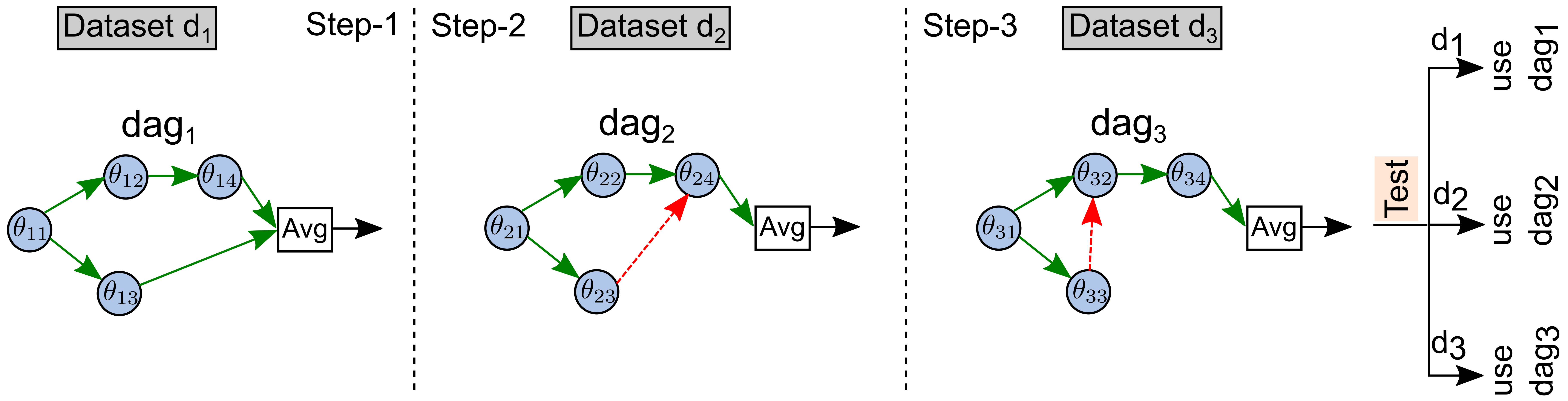


Figure: Continual architecture search (CAS) approach: green, solid edges (weight parameters) are shared, newly-learned edges are represented with red, dashed edges.

Generalizable NAS on Sequential Tasks

CAS Results on Text Classification

Models	QNLI	RTE	WNLI
PREVIOUS WORK			
BiLSTM+ELMo (2018)	69.4	50.1	65.1
BiLSTM+ELMo+Attn (2018)	61.1	50.3	65.1
BASELINES			
Baseline (with ELMo)	73.2	52.3	65.1
ENAS (Architecture Search)	74.5	52.9	65.1

Table: Test results on GLUE tasks for various models: Baseline, ENAS, and CAS (continual architecture search). The CAS results maintain statistical equality across each step.

[Dagan et al., 2006; Bar Haim etal., 2006; Giampiccoloetal., 2007; Bentivoglietal., 2009; Levesque et al., 2011; Rajpurkar et al. 2016; Wang et al., 2018]

CAS Results on Text Classification

Models	QNLI	RTE	WNLI
PREVIOUS WORK			
BiLSTM+ELMo (2018)	69.4	50.1	65.1
BiLSTM+ELMo+Attn (2018)	61.1	50.3	65.1
BASELINES			
Baseline (with ELMo)	73.2	52.3	65.1
ENAS (Architecture Search)	74.5	52.9	65.1
CAS RESULTS			
CAS Step-1 (QNLI training)	73.8	N/A	N/A
CAS Step-2 (RTE training)	73.6	54.1	N/A
CAS Step-3 (WNLI training)	73.3	54.0	64.4

Table: Test results on GLUE tasks for various models: Baseline, ENAS, and CAS (continual architecture search). The CAS results maintain statistical equality across each step.

[Dagan et al., 2006; Bar Haim etal., 2006; Giampiccoloetal., 2007; Bentivoglietal., 2009; Levesque et al., 2011; Rajpurkar et al. 2016; Wang et al., 2018]

CAS Results on Text Classification

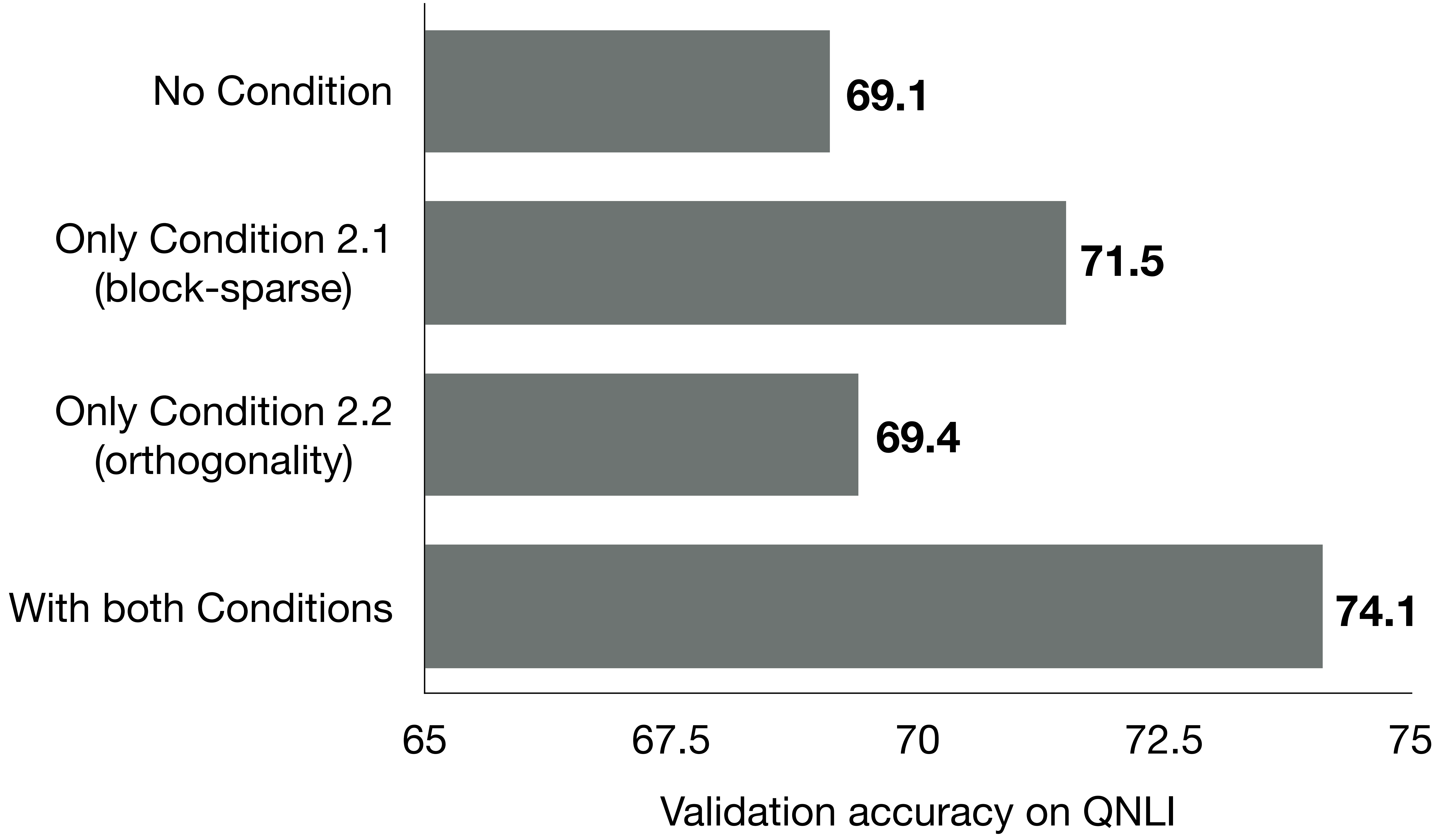
Models	QNLI	RTE	WNLI
PREVIOUS WORK			
BiLSTM+ELMo (2018)	69.4	50.1	65.1
BiLSTM+ELMo+Attn (2018)	61.1	50.3	65.1
BASELINES			
Baseline (with ELMo)	73.2	52.3	65.1
ENAS (Architecture Search)	74.5	52.9	65.1
CAS RESULTS			
CAS Step-1 (QNLI training)	73.8	N/A	N/A
CAS Step-2 (RTE training)	73.6	54.1	N/A
CAS Step-3 (WNLI training)	73.3	54.0	64.4

Difference is statistically insignificant, hence CAS is maintaining performance sequentially

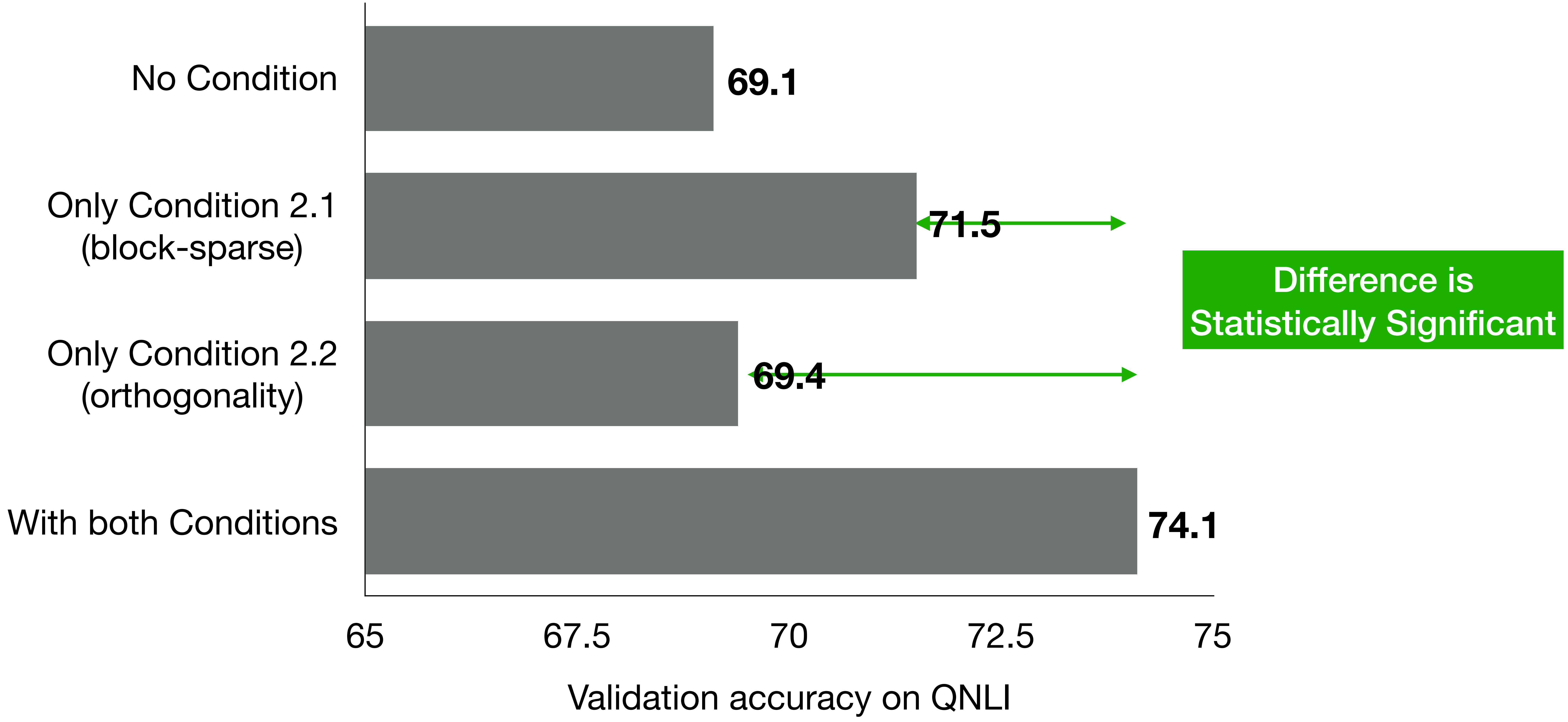
Table: Test results on GLUE tasks for various models: Baseline, ENAS, and CAS (continual architecture search). The CAS results maintain statistical equality across each step.

[Dagan et al., 2006; Bar Haim etal., 2006; Giampiccoloetal., 2007; Bentivoglietal., 2009; Levesque et al., 2011; Rajpurkar et al. 2016; Wang et al., 2018]

CAS Ablation



CAS Ablation



CAS Results on Video Captioning

Models	MSR-VTT					MSVD				
	C	B	R	M	AVG	C	B	R	M	AVG
Baseline (Pasunuru and Bansal, 2017b)	48.2	40.8	60.7	28.1	44.5	85.8	52.5	71.2	35.0	61.1
ENAS	48.9	41.3	61.2	28.1	44.9	87.2	52.9	71.7	35.2	61.8
CAS Step-1 (MSR-VTT training)	48.9	41.1	60.5	27.5	44.5	N/A	N/A	N/A	N/A	N/A
CAS Step-2 (MSVD training)	48.4	40.1	59.9	27.1	43.9	88.1	52.4	71.3	35.1	61.7

Table: Video captioning results with Baseline, ENAS, and CAS.

[Chen & Dolan, 2011; Xu et al., 2016]

CAS Results on Video Captioning

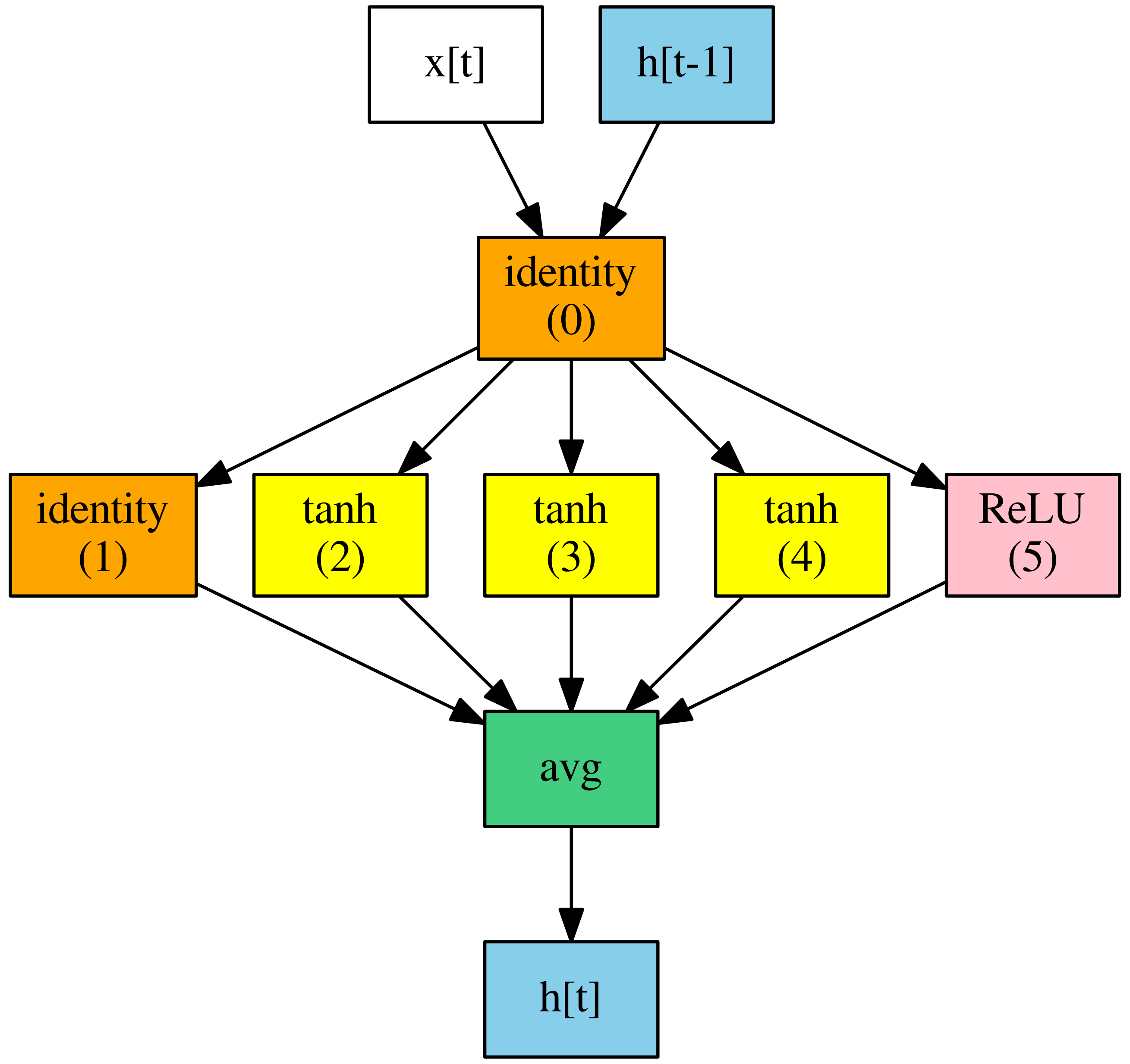
Models	MSR-VTT					MSVD				
	C	B	R	M	AVG	C	B	R	M	AVG
Baseline (Pasunuru and Bansal, 2017b)	48.2	40.8	60.7	28.1	44.5	85.8	52.5	71.2	35.0	61.1
ENAS	48.9	41.3	61.2	28.1	44.9	87.2	52.9	71.7	35.2	61.8
CAS Step-1 (MSR-VTT training)	48.9	41.1	60.5	27.5	44.5	N/A	N/A	N/A	N/A	N/A
CAS Step-2 (MSVD training)	48.4	40.1	59.9	27.1	43.9	88.1	52.4	71.3	35.1	61.7

Table: Video captioning results with Baseline, ENAS, and CAS.

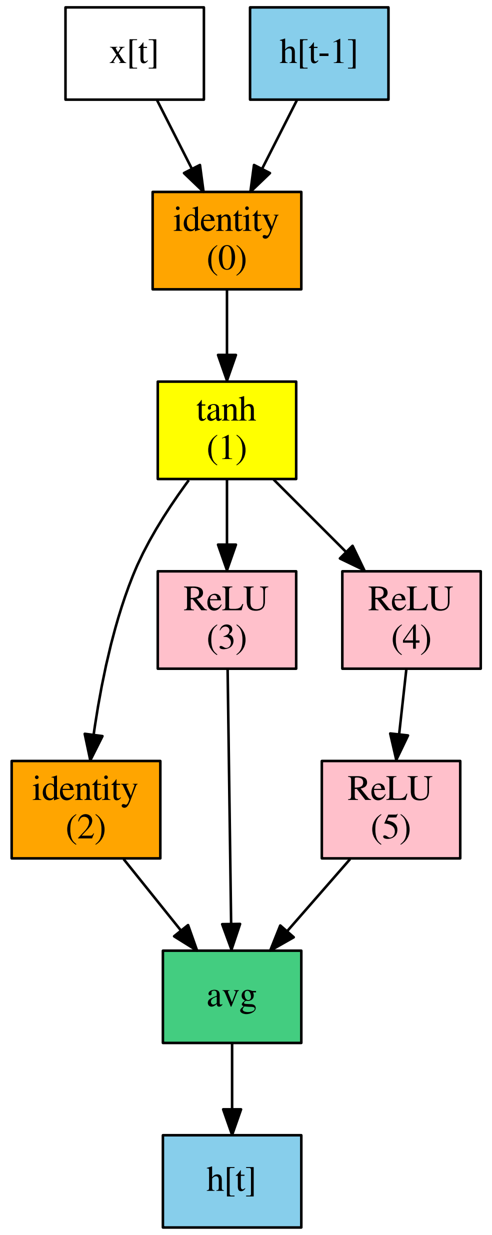
Difference is Statistically insignificant (also based on Human evaluation)
Hence CAS is maintaining performance sequentially

[Chen & Dolan, 2011; Xu et al., 2016]

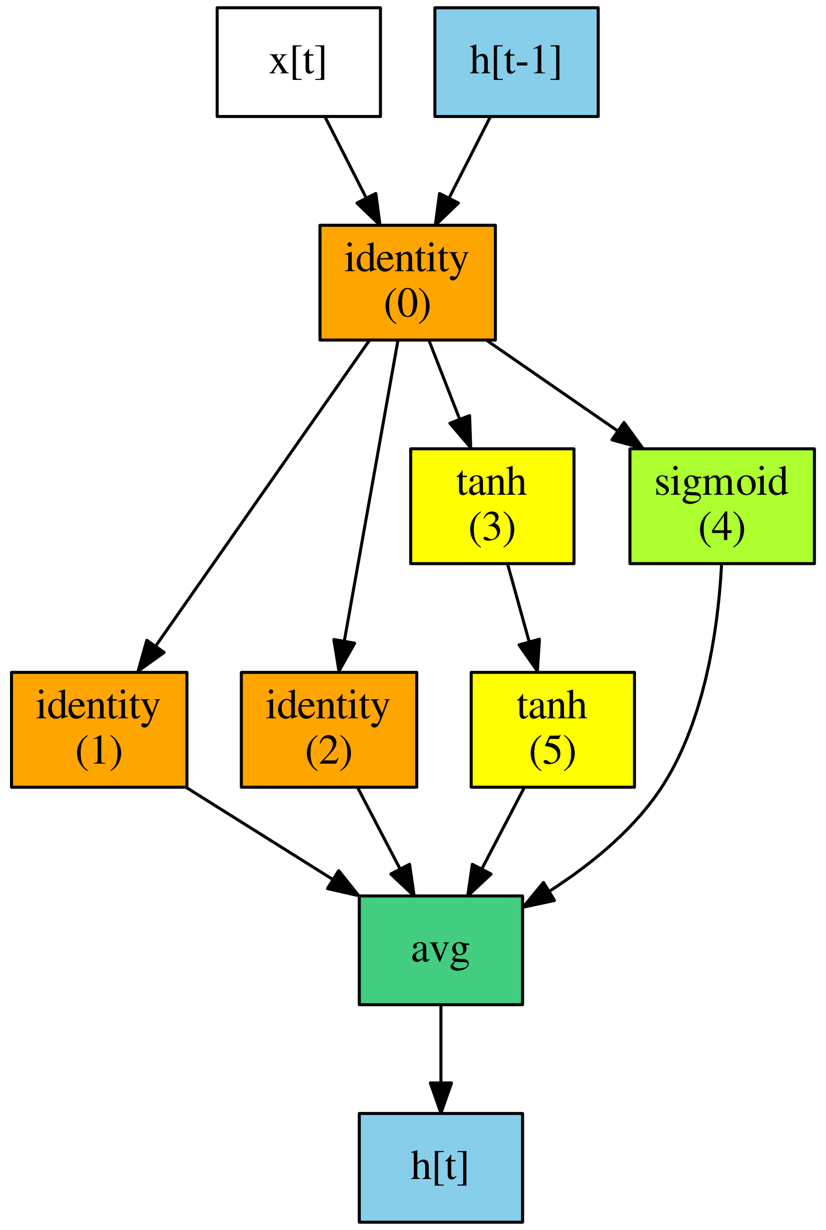
CAS Learned Cells



(a) Step-1



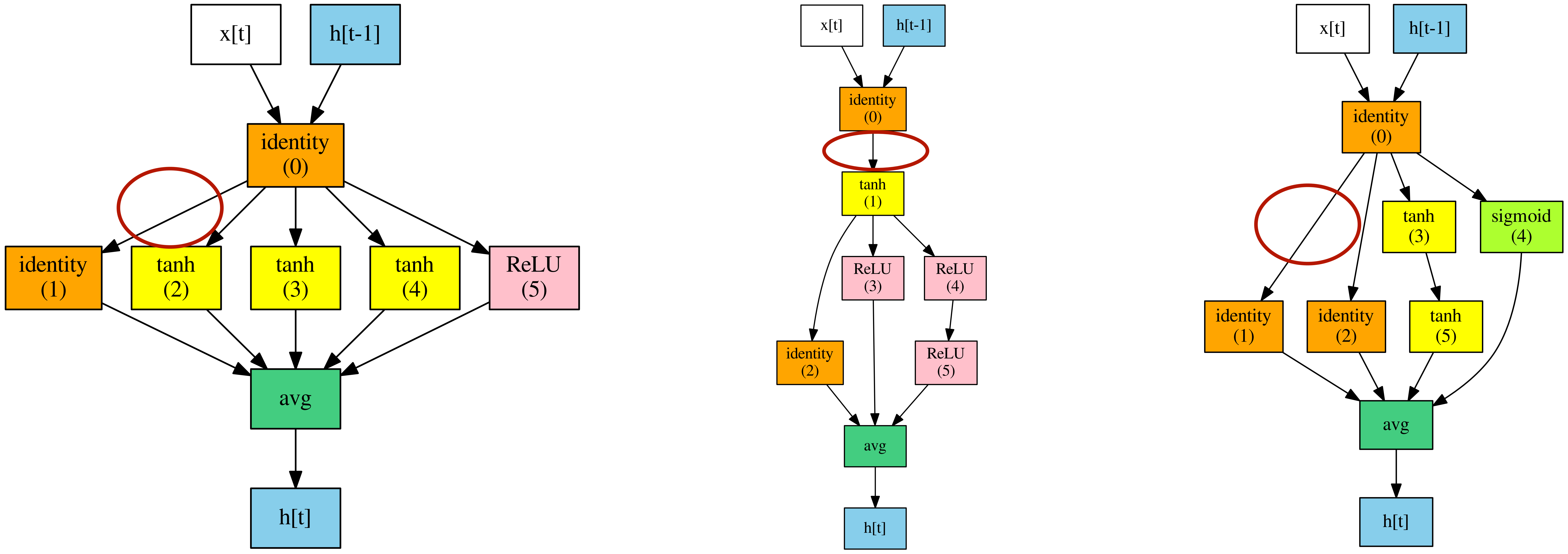
(b) Step-2



(c) Step-3

Figure: Learned cell structures for step-1, step-2, and step-3 of continual architecture search for GLUE tasks.

CAS Learned Cells

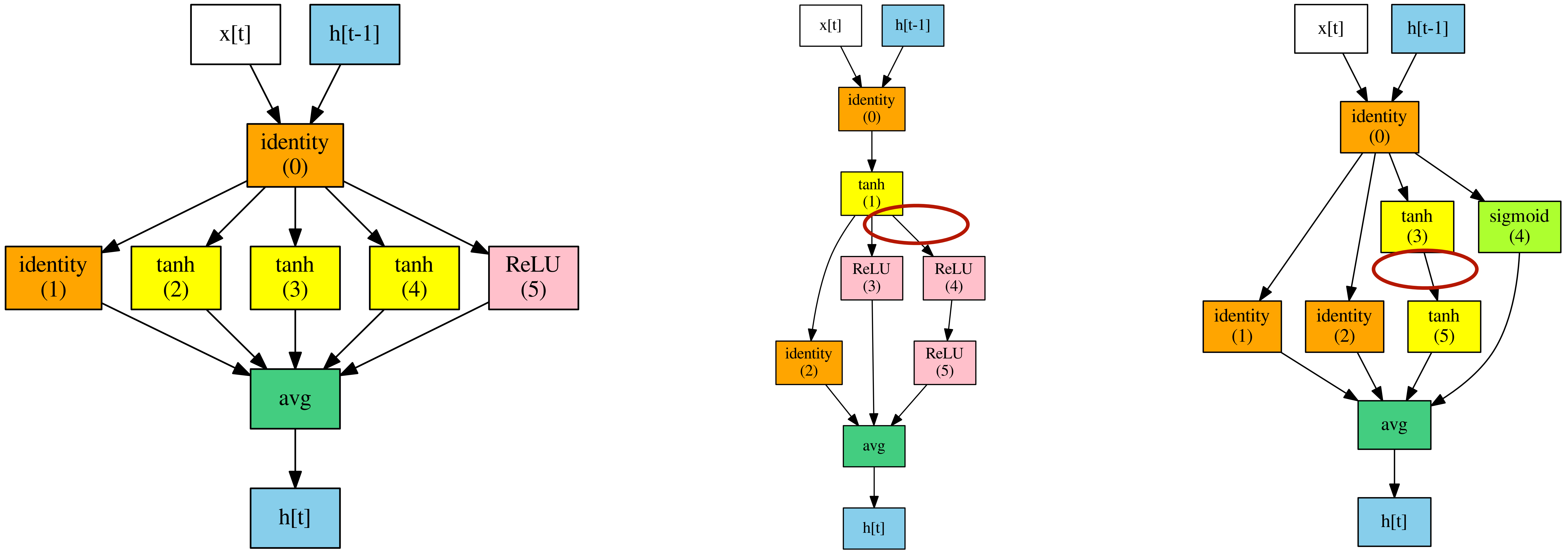


(a) Step-1 (b) Step-2 (c) Step-3

Share common edges

Figure: Learned cell structures for step-1, step-2, and step-3 of continual architecture search for GLUE tasks.

CAS Learned Cells



(a) Step-1 (b) Step-2 (c) Step-3

Task-specific new edges

Figure: Learned cell structures for step-1, step-2, and step-3 of continual architecture search for GLUE tasks.

Generalizable Cell on Multiple Tasks

- Architectures found by *NAS* are dataset dependent
- Human designed cell (e.g., *LSTM* and *GRU*) work well across multiple datasets

Generalizable Cell on Multiple Tasks

- Architectures found by *NAS* are dataset dependent
- Human designed cell (e.g., *LSTM* and *GRU*) work well across multiple datasets

Can we learn generalizable NAS cell structures?

Generalizable Cell on Multiple Tasks

- Architectures found by *NAS* are dataset dependent
- Human designed cell (e.g., *LSTM* and *GRU*) work well across multiple datasets

Can we learn generalizable NAS cell structures?

Multi-Task Learning!!

Multi-Task Architecture Search (MAS)

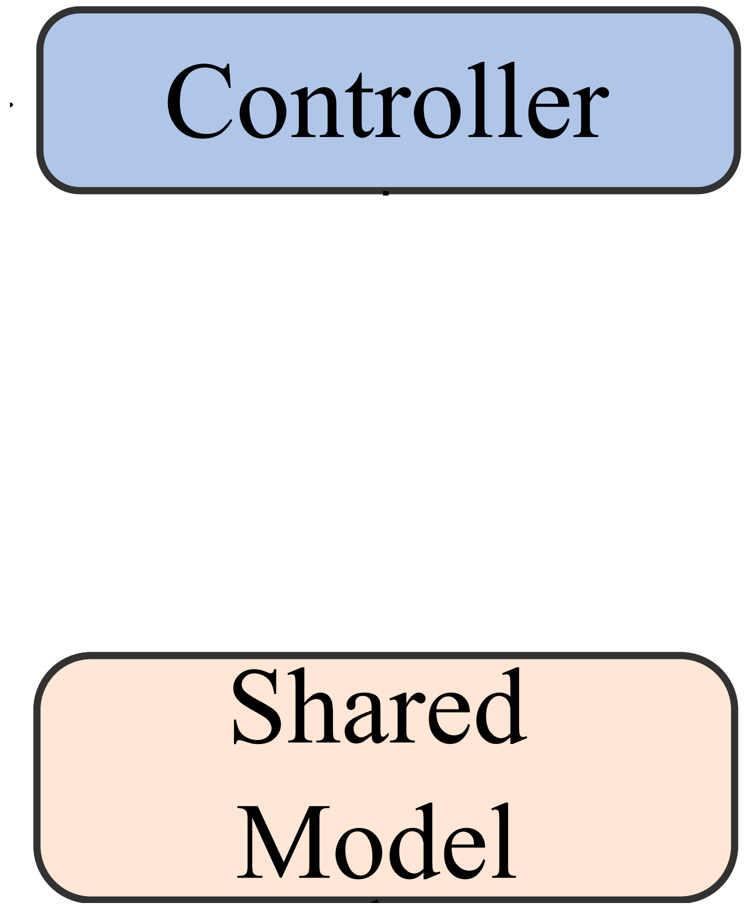


Figure: Multi-task cell structure learning using joint rewards from n datasets.

Multi-Task Architecture Search (MAS)

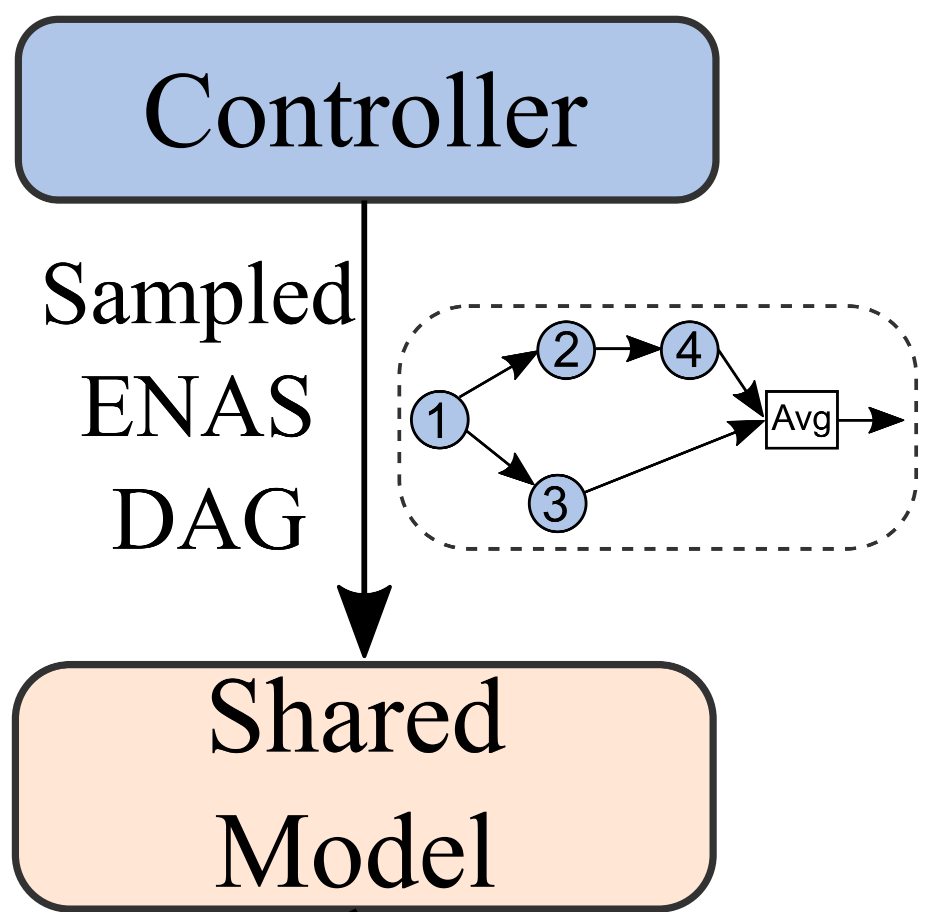


Figure: Multi-task cell structure learning using joint rewards from n datasets.

Multi-Task Architecture Search (MAS)

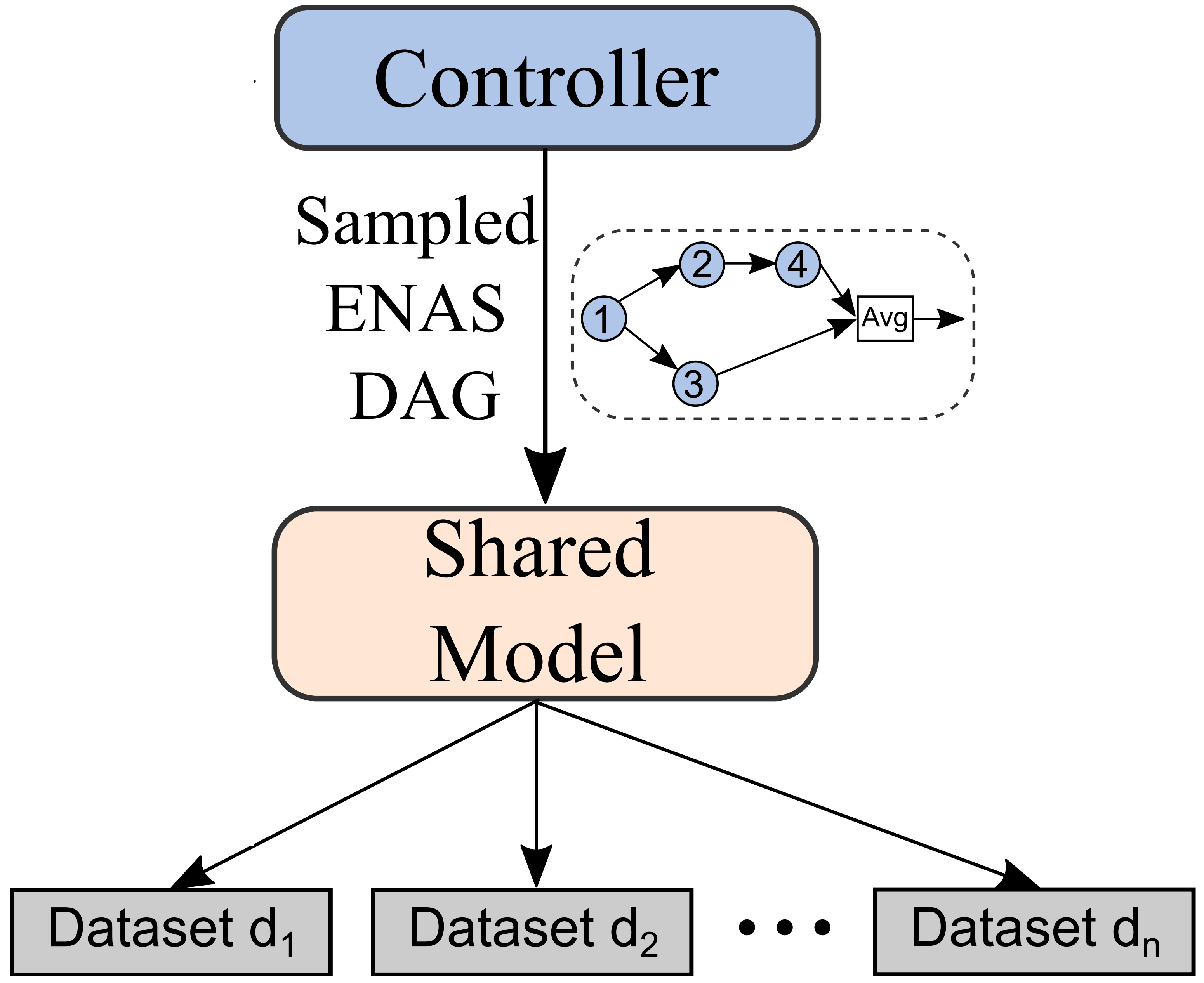


Figure: Multi-task cell structure learning using joint rewards from n datasets.

Multi-Task Architecture Search (MAS)

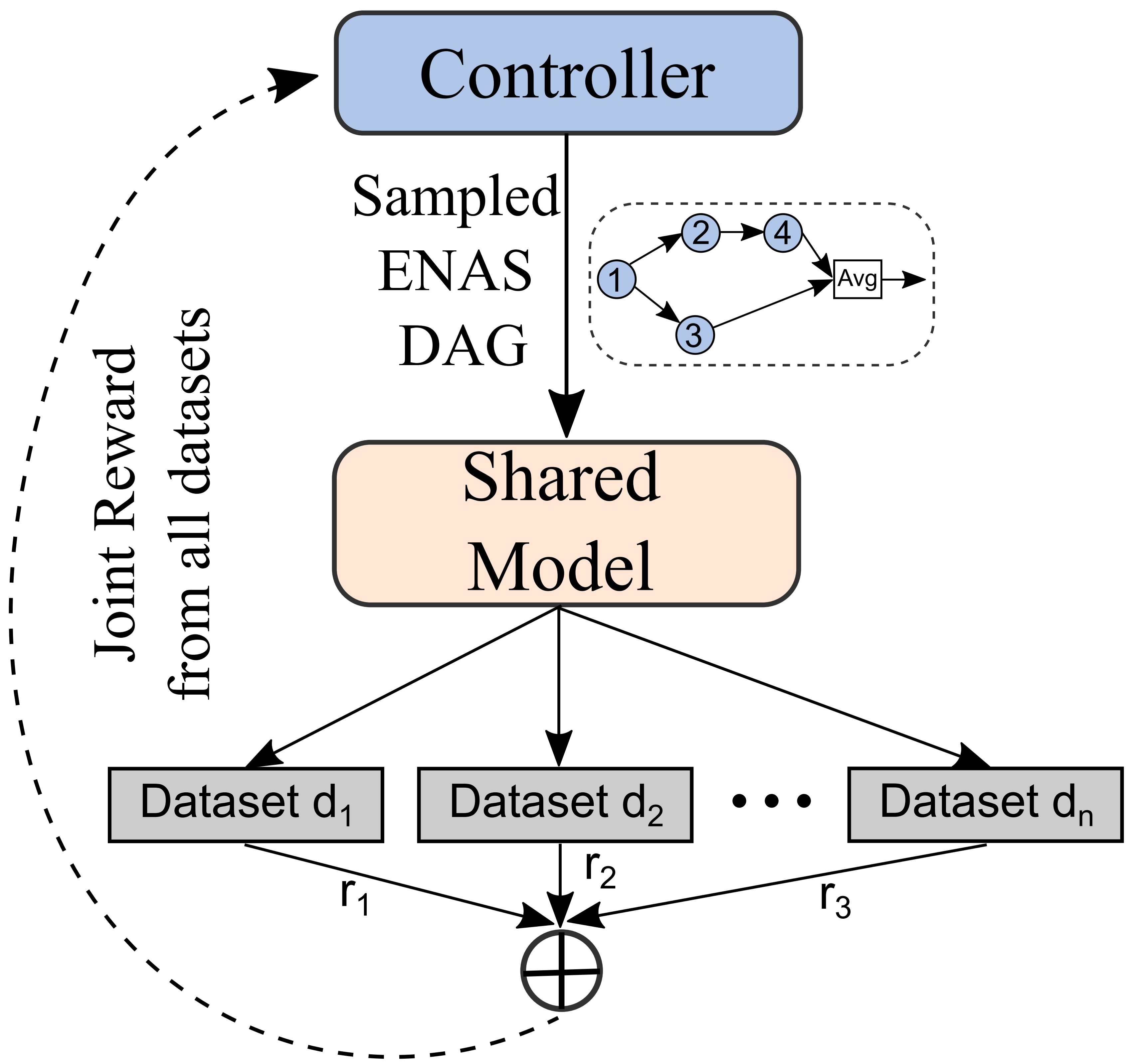
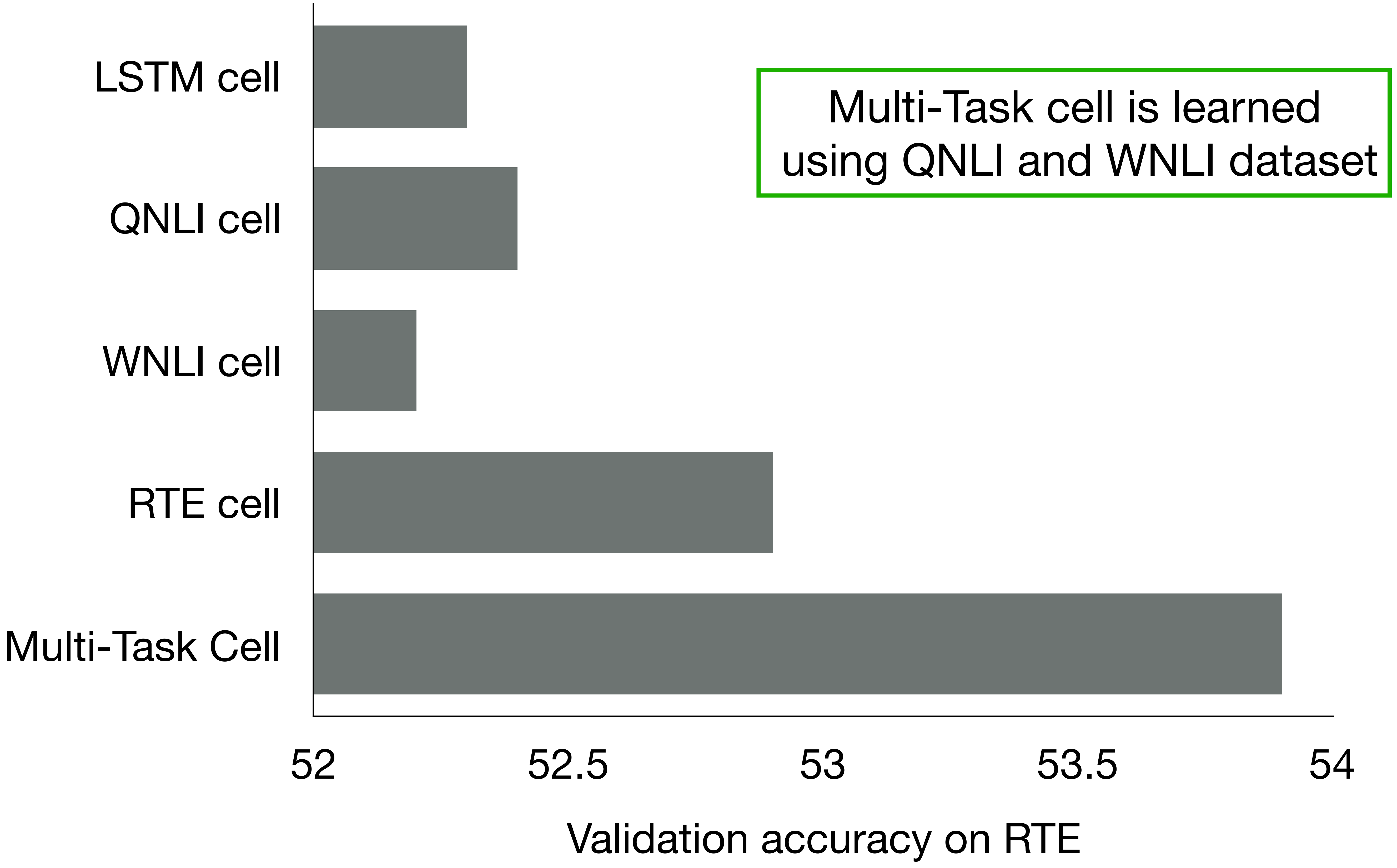
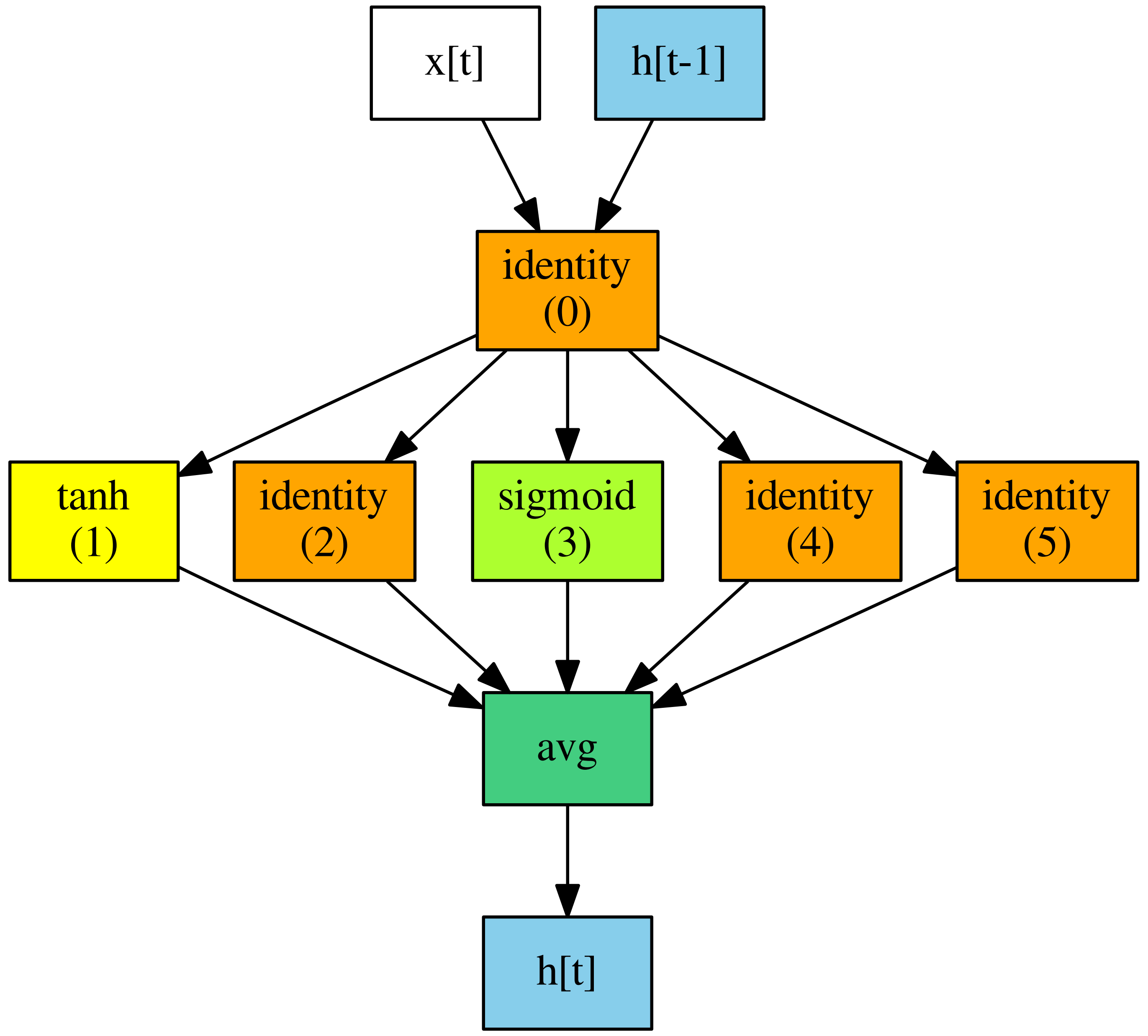


Figure: Multi-task cell structure learning using joint rewards from n datasets.

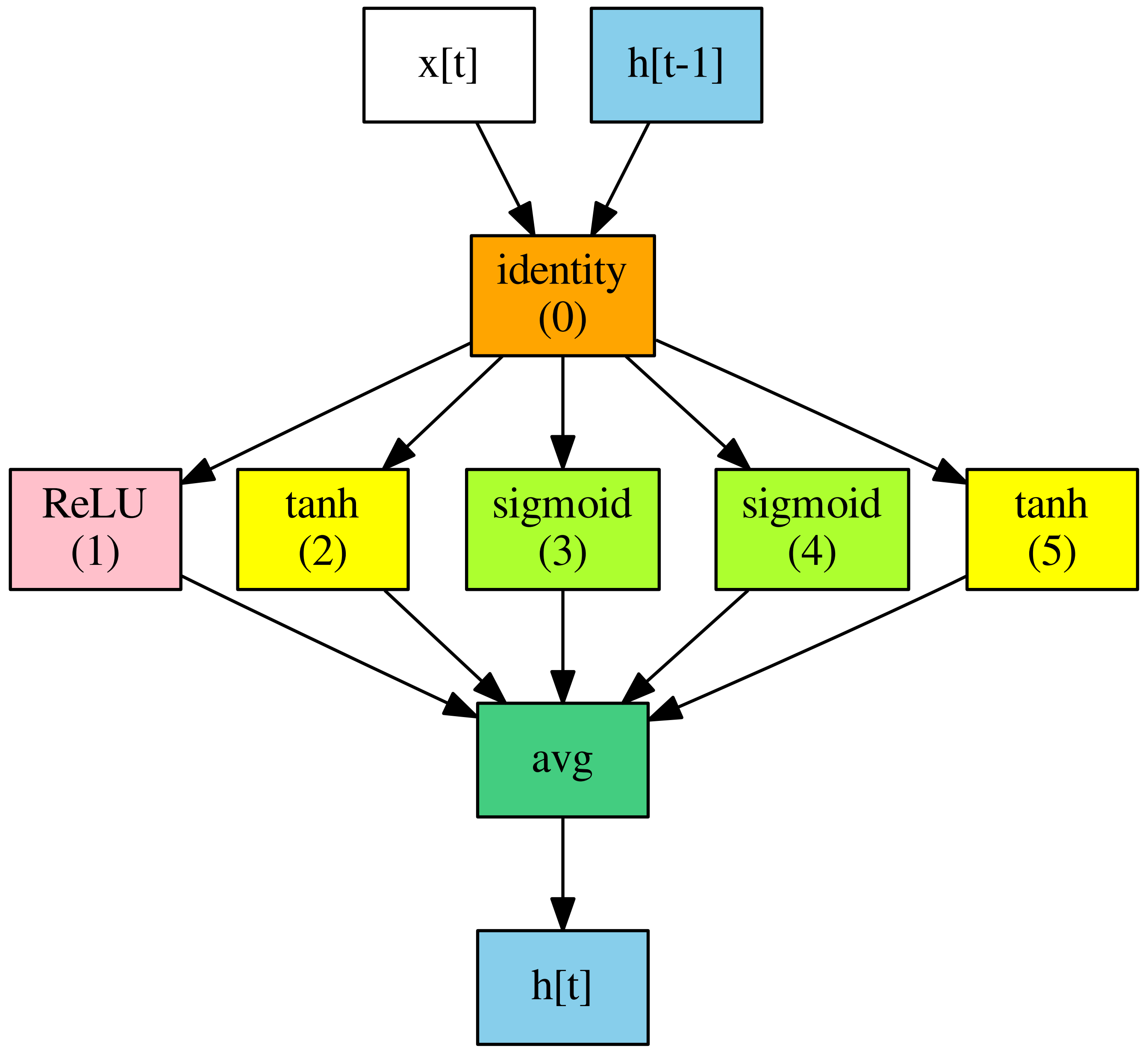
MAS Results on Text Classification



MAS Learned Cells



(a) MAS cell

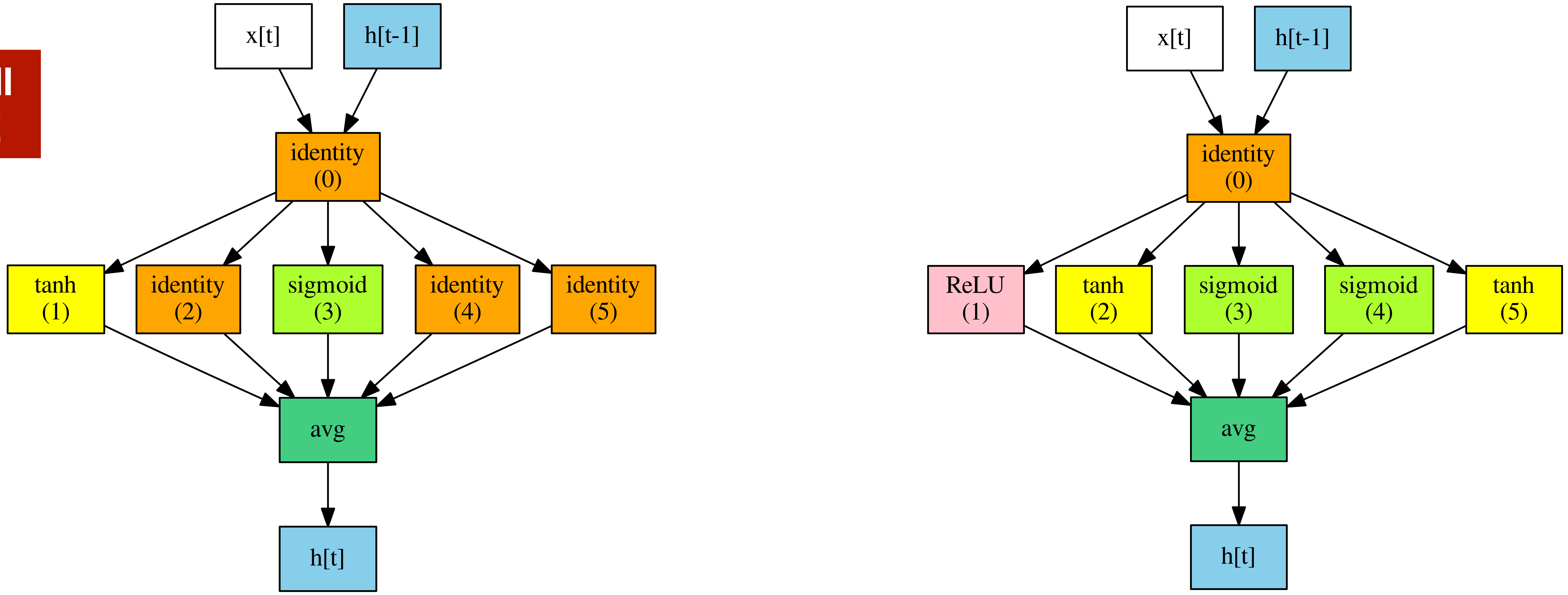


(b) RTE cell

Figure: Learned Multi-task and RTE cell Structures.

MAS Learned Cells

MAS-learned cell is less complex!



(a) MAS cell

(b) RTE cell

Figure: Learned Multi-task and RTE cell Structures.

Thanks!

Ramkanth Pasunuru

www.rama-kanth.com

Mohit Bansal

www.cs.unc.edu/~mbansal/

Code: https://github.com/ramakanth-pasunuru/CAS-MAS

Acknowledgements: We thank the reviewers for their helpful comments. This work was supported by DARPA (YFA17-D17AP00022), and faculty awards from Google, Facebook, and Salesforce.