

Flowformer: Linearizing Transformers with Conservation Flows

Haixu Wu¹ Jialong Wu¹ Jiehui Xu¹ Jianmin Wang¹ Mingsheng Long¹



Haixu Wu



Jialong Wu



Jiehui Xu

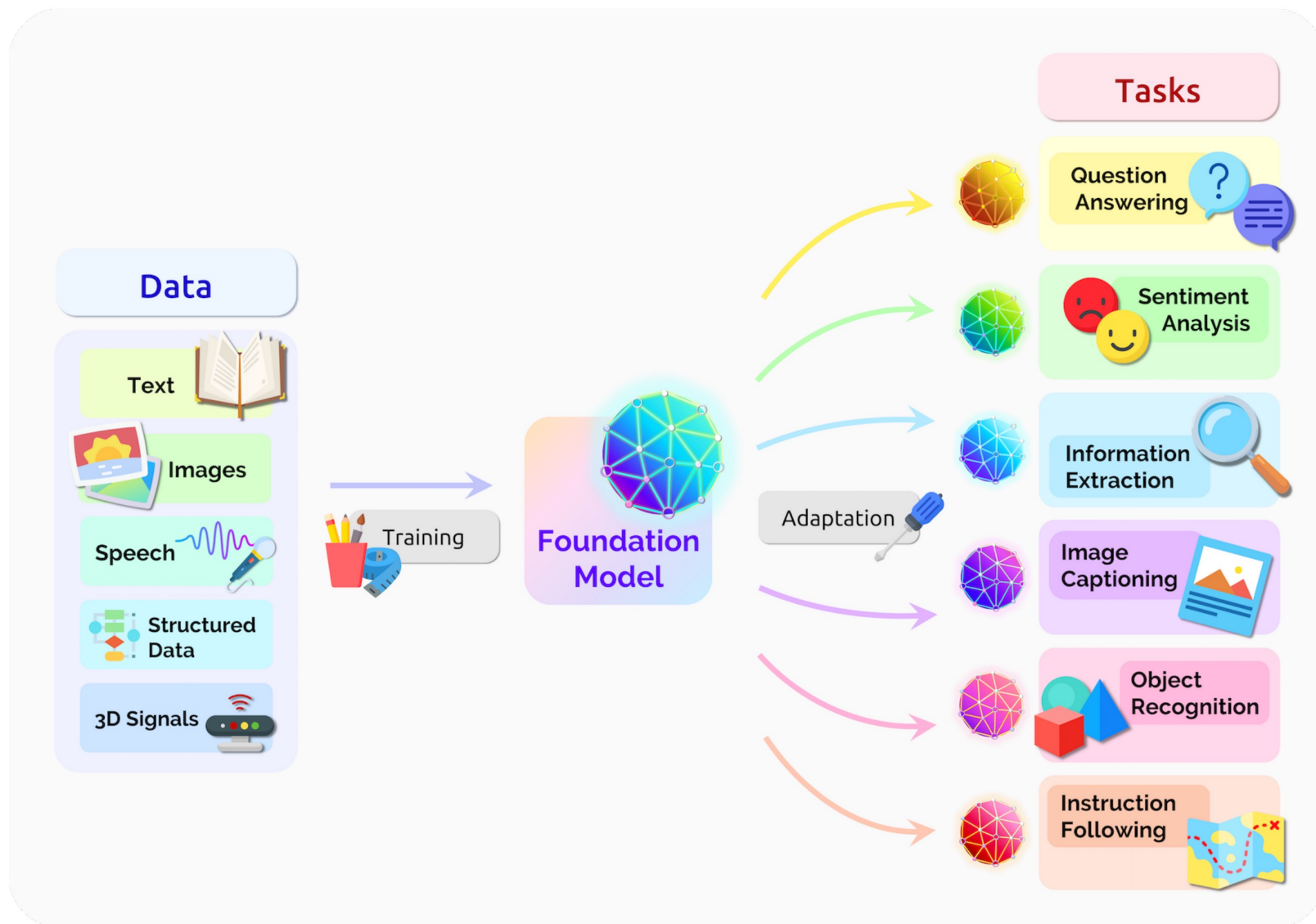


Jianmin Wang



Mingsheng Long

Foundation Models



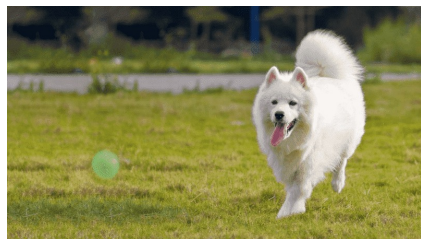
[Data Universal]

Learn from various modalities

[Task Universal]

Adapt to a wide range of
downstream tasks

A Universal Architecture for General Proposes



Image



Language



Time
Series



Agent
Trajectory

**Universal
Architecture**

A Universal Architecture for General Proposes



Image



Language

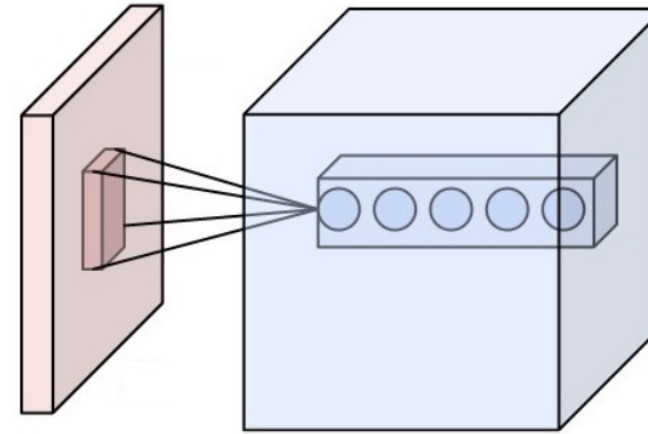


Time Series



Agent Trajectory

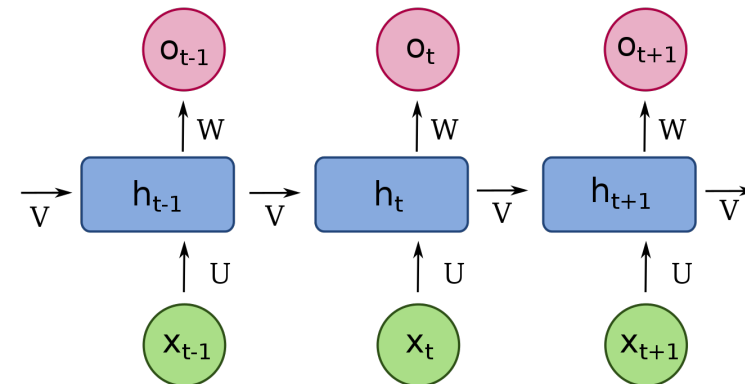
Universal Architecture



CNNs?

Locality

Shift Invariance ☹️



RNNs?

Markov ☹️

A Universal Architecture for General Proposes



Image



Language

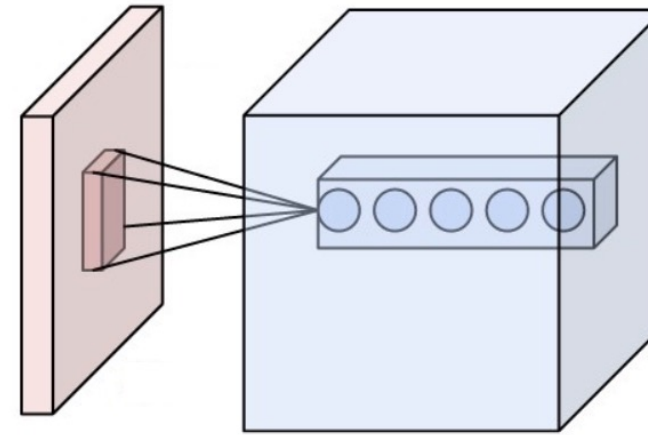


Time Series



Agent Trajectory

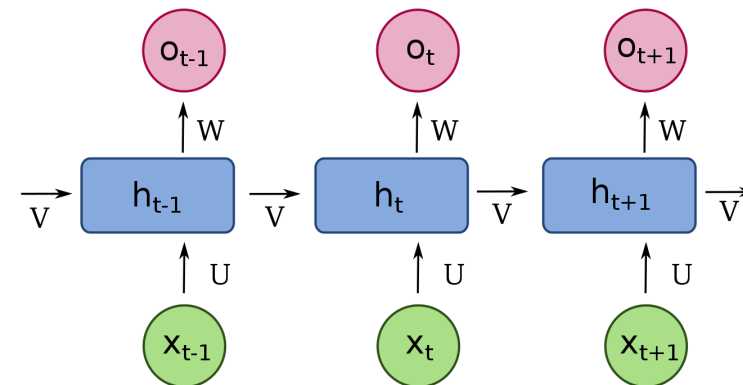
Universal Architecture



CNNs?

Locality

Shift Invariance ☹️

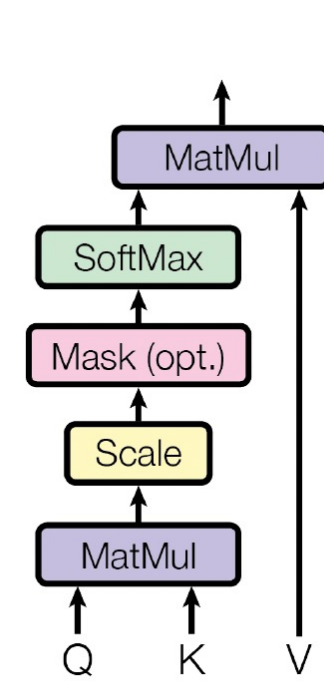
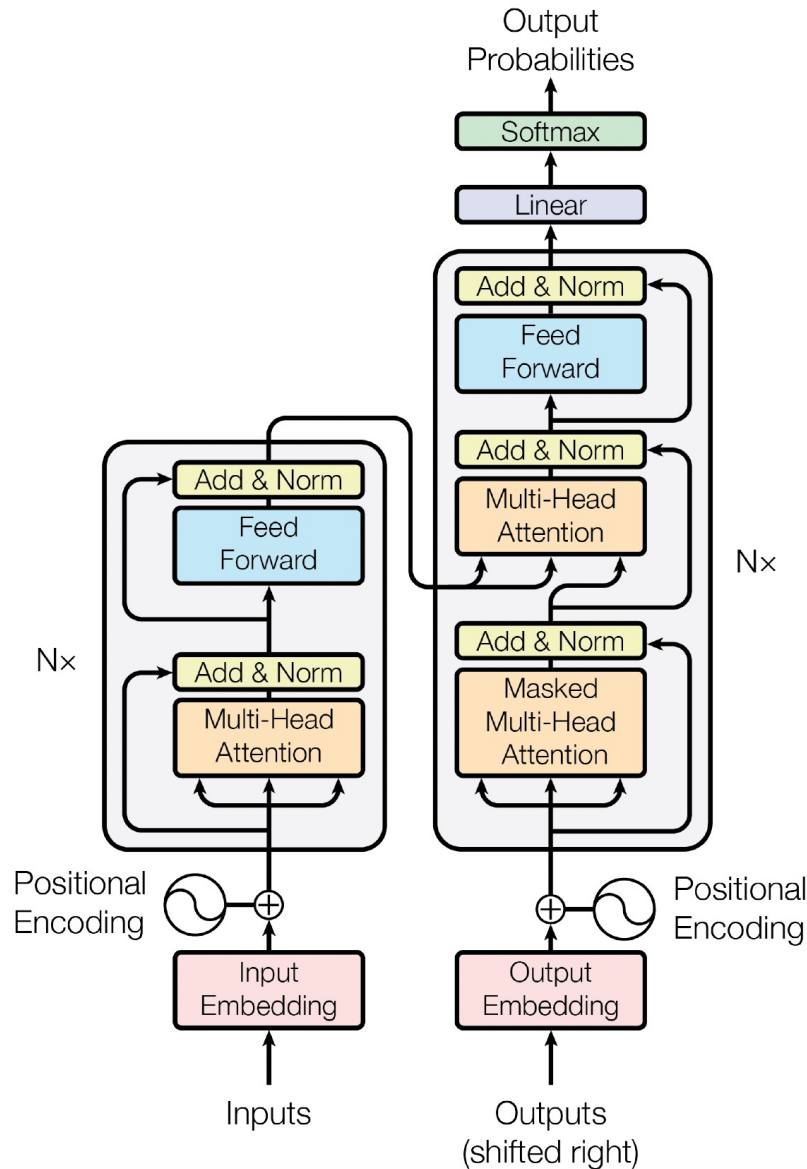


RNNs?

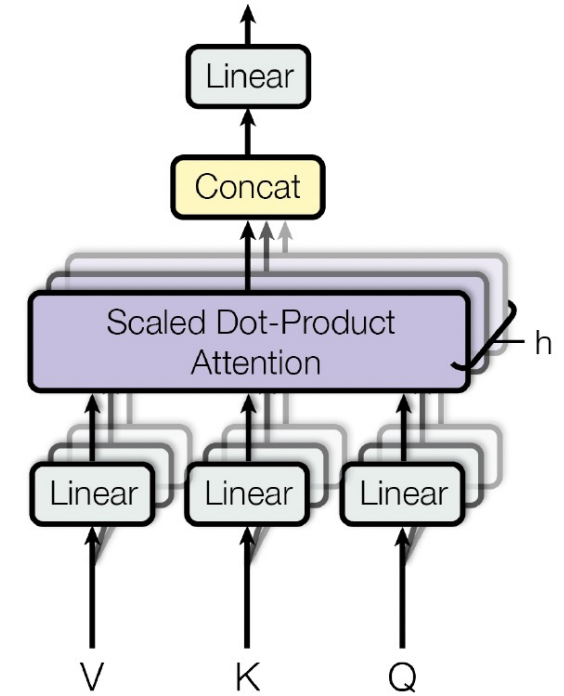
Markov ☹️

Specific Inductive Biases Limit the Model Universality

Transformers



Self-Attention

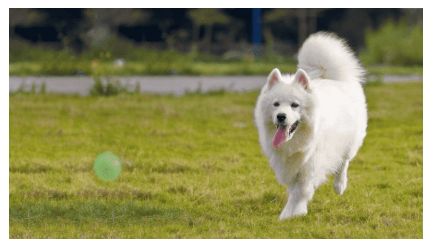


Multi-head **Self-Attention**

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Dot-product Similarity & Without Specific Inductive Biases

General Relation Modeling



Image



Relation among **Image Patches**



Language



Relation among **Words**

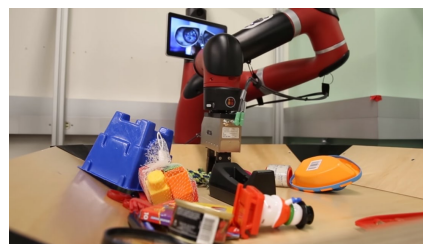
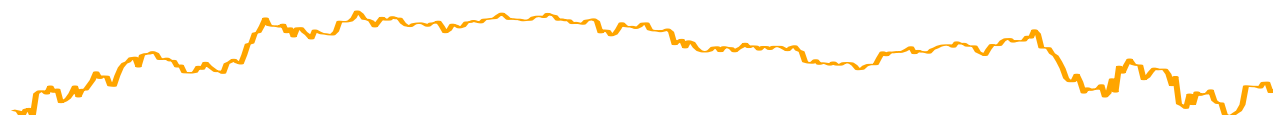
[SOS] Flowformer is a **task-universal** linear Transformer. [EOS]



Time
Series



Relation among **Time Points**



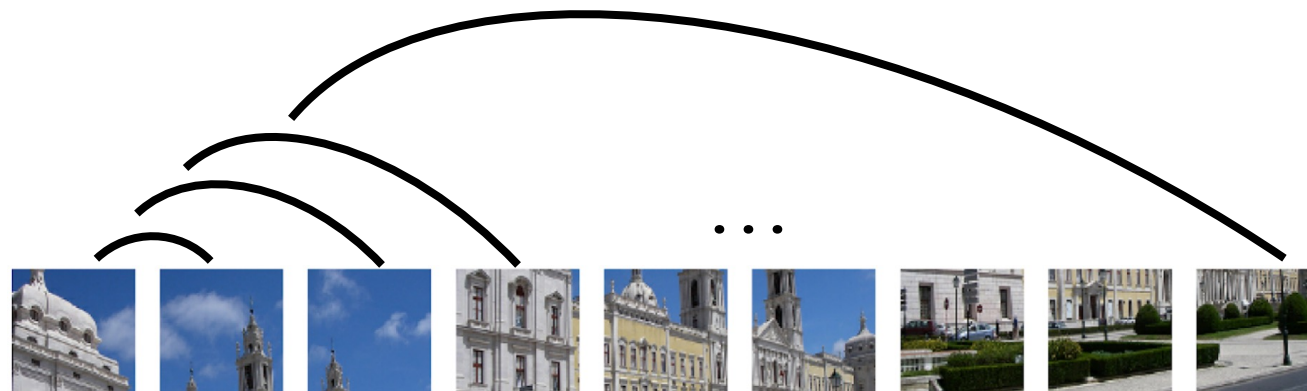
Agent
Trajectory



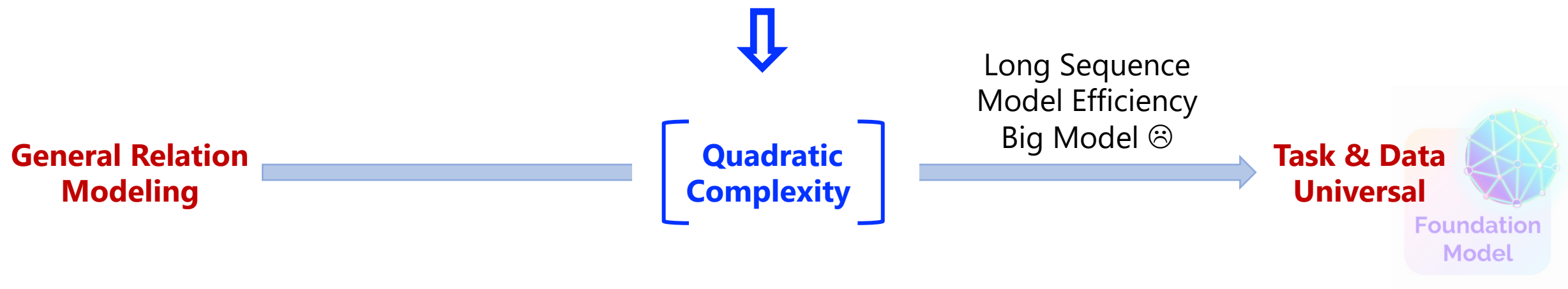
Relation among **Agent-Environment Interactions**



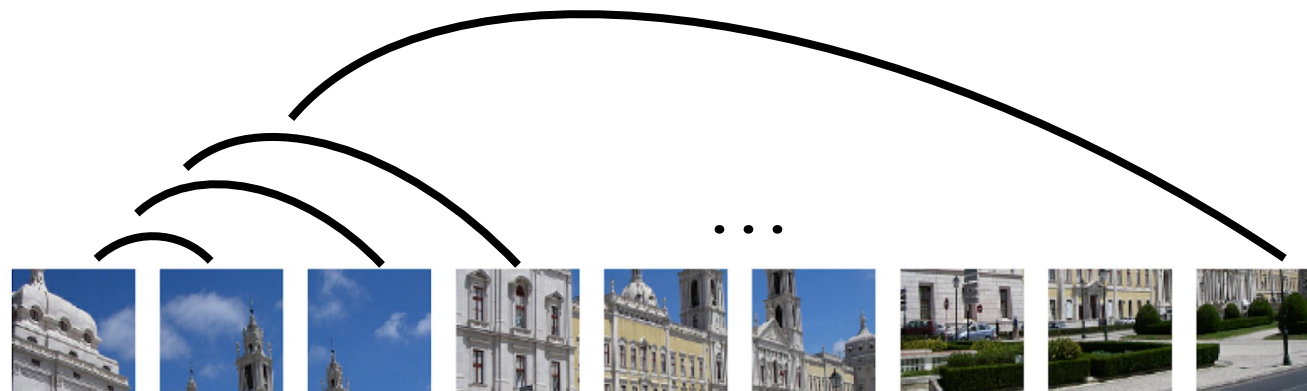
Quadratic Complexity in Self-Attention



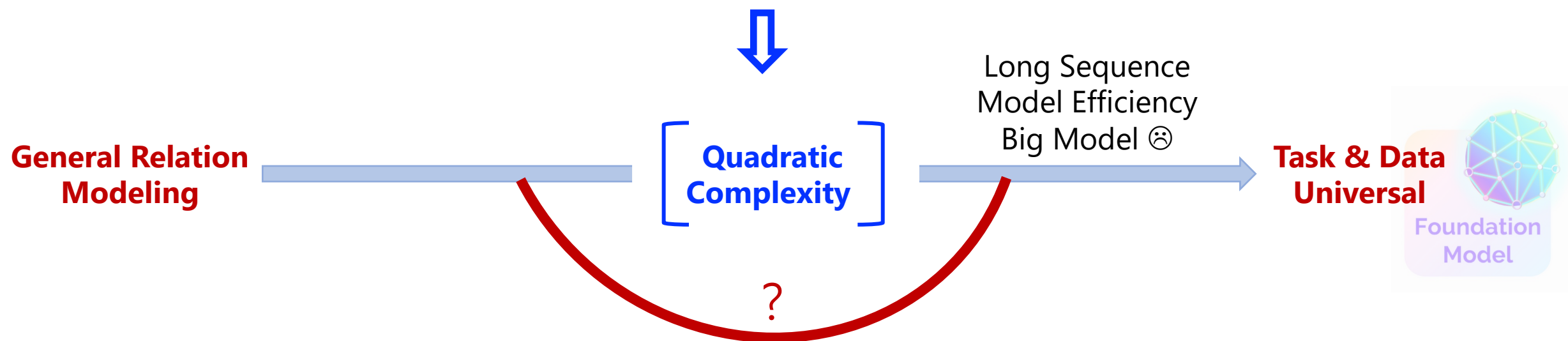
Pair-wise Relation Modeling: $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$



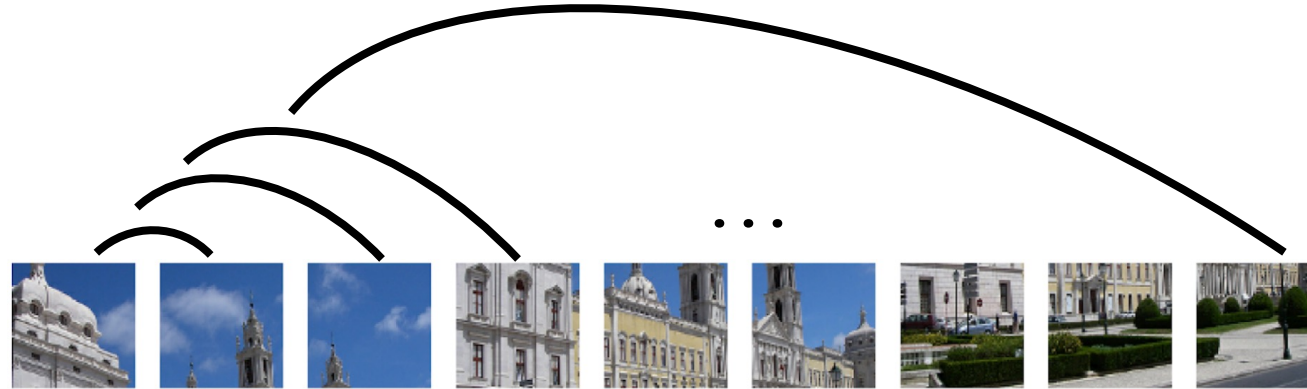
Quadratic Complexity in Self-Attention



Pair-wise Relation Modeling: $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$



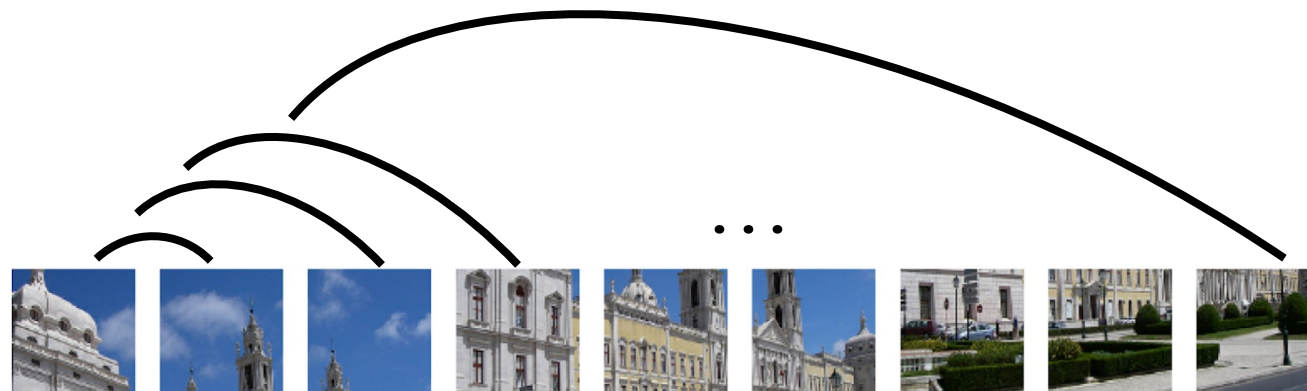
Quadratic Complexity in Self-Attention



Pair-wise Relation Modeling: $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$

$\mathcal{O}(n^2 d)$

Quadratic Complexity in Self-Attention



Pair-wise Relation Modeling: $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$

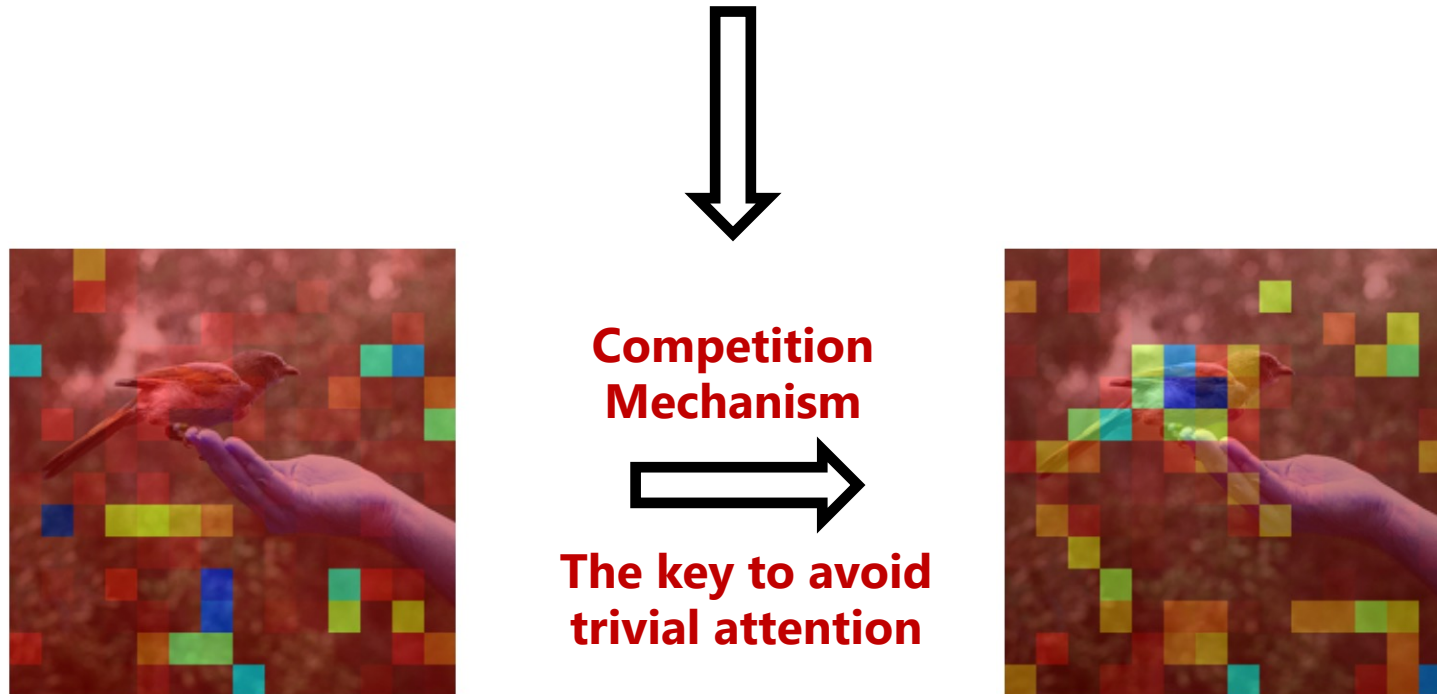
$\mathcal{O}(n^2 d)$

Can we remove Softmax function?

$$(QK^T)V = Q(K^TV) \Rightarrow \mathcal{O}(n^2 d) \rightarrow \mathcal{O}(nd^2)$$

Recap: Softmax Function

Softmax function is proposed as a differentiable generalization of the ***"winner-take-all"*** picking maximum operation.



Bridle et al. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *NeurIPS 1989*.

Recap: Softmax Function

Softmax function is proposed as a differentiable generalization of the ***"winner-take-all"*** picking maximum operation.

$$\begin{array}{ccc} \phi(Q)(\phi(K)^T V) & \longleftrightarrow & \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V \\ + & & \\ \text{Competition Mechanism} & & \end{array}$$

Bridle et al. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *NeurIPS 1989*.

Recap: Softmax Function

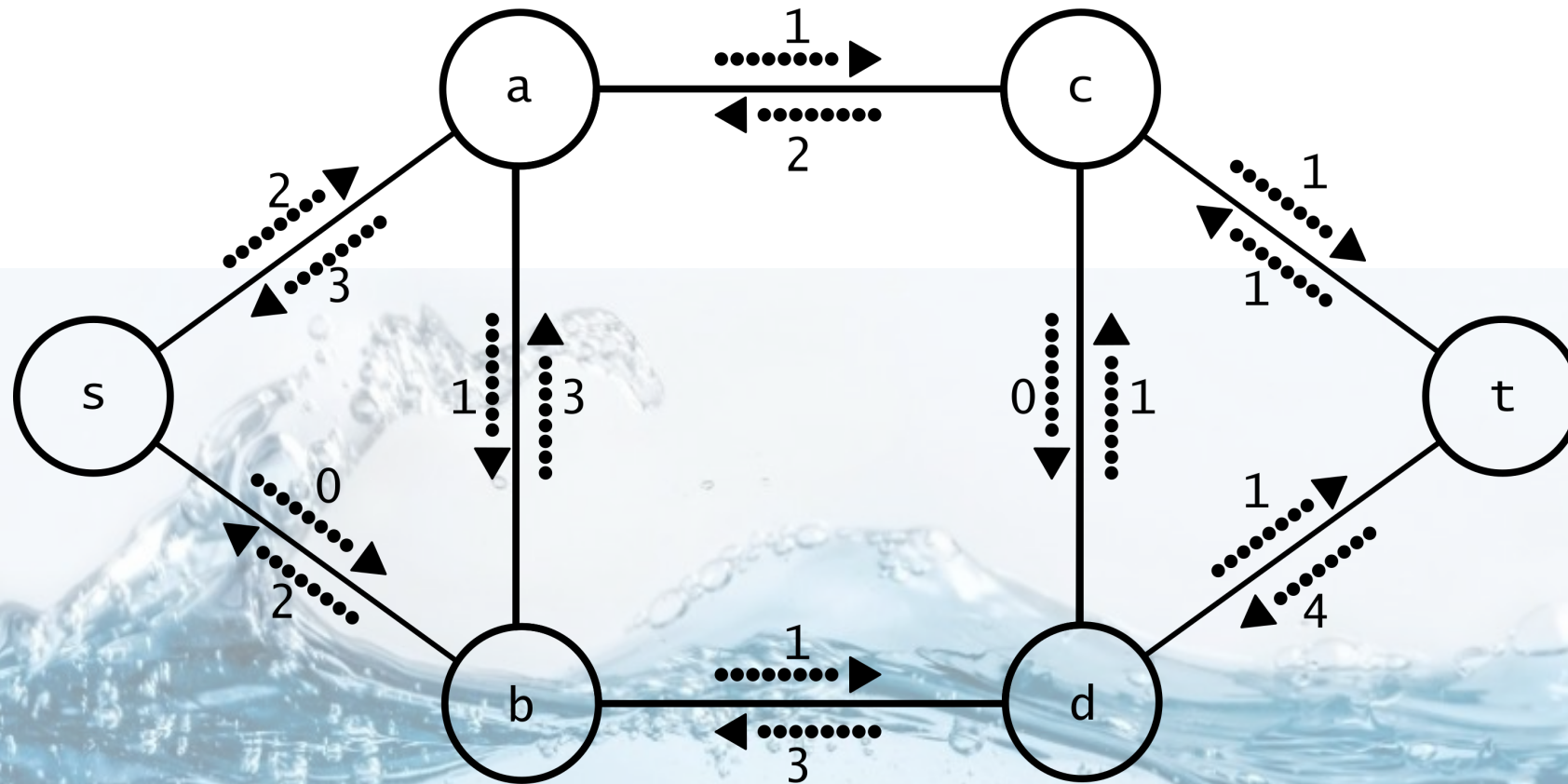
Softmax function is proposed as a differentiable generalization of the ***"winner-take-all"*** picking maximum operation.

$$\begin{array}{ccc} \phi(Q)(\phi(K)^T V) & \longleftrightarrow & \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V \\ + & & \\ \text{Competition Mechanism} & & \end{array}$$

"fixed resource will cause competition"

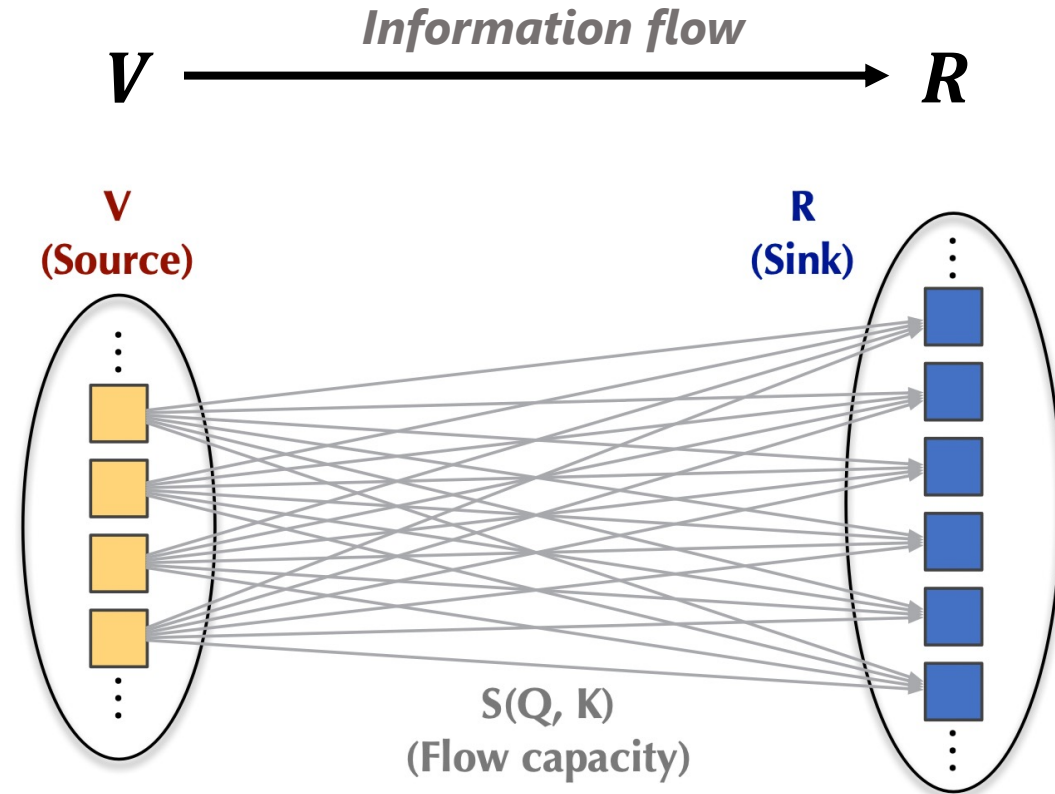
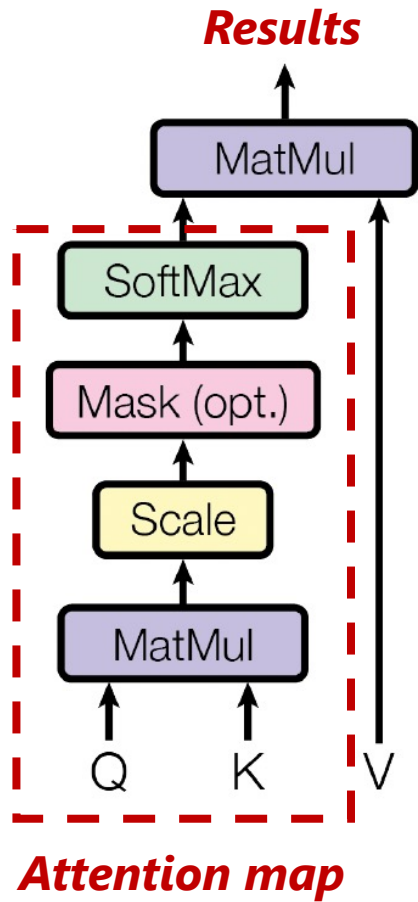
Bridle et al. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *NeurIPS 1989*.

Flow Network Theory



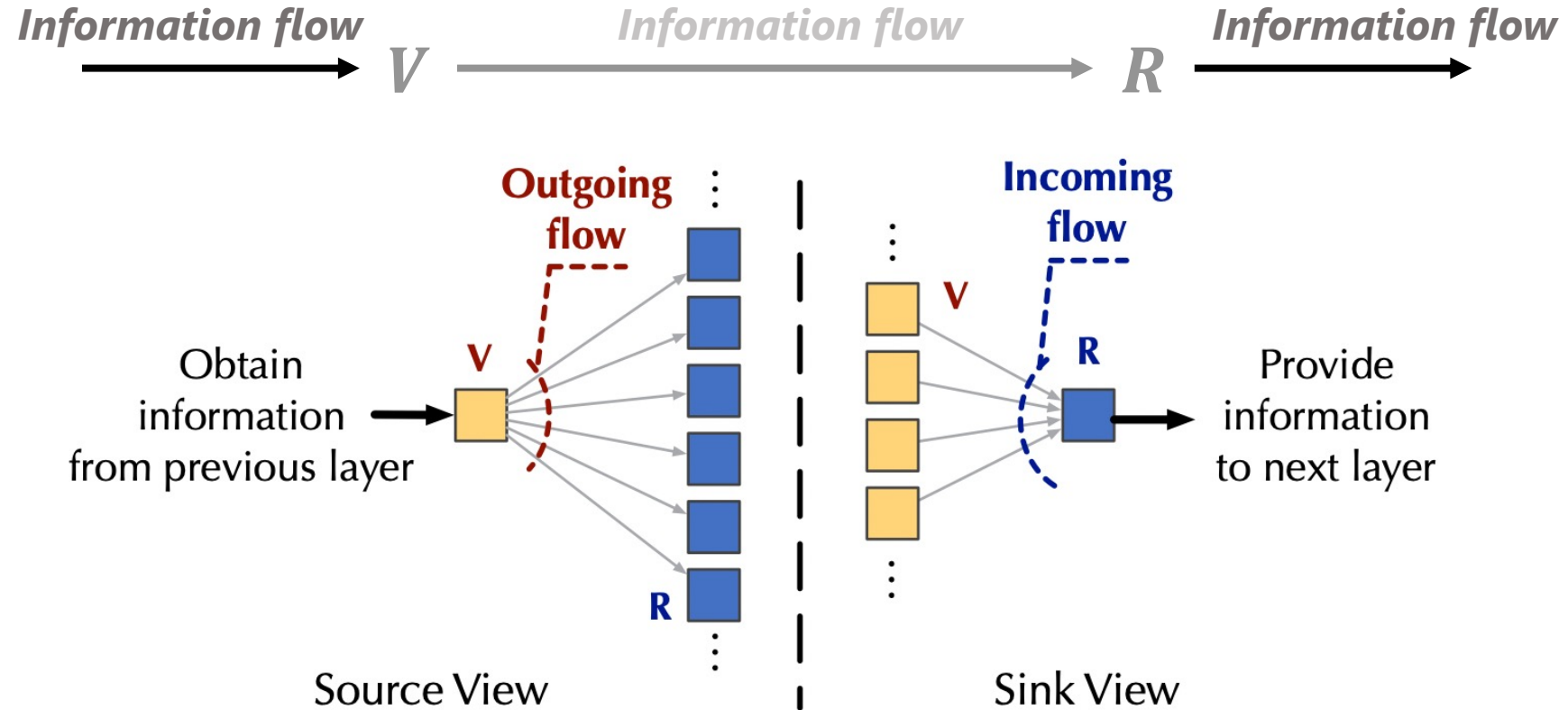
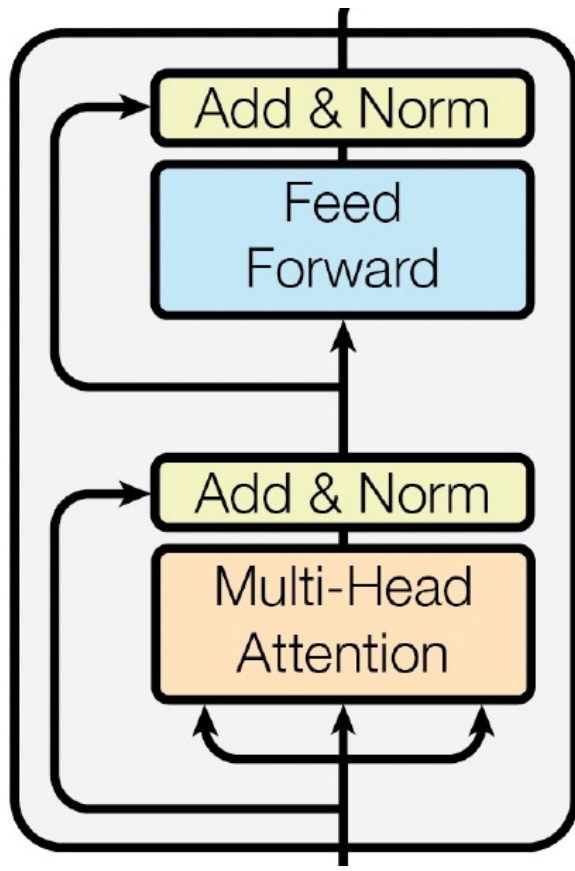
[Conservation Property]: The incoming flow capacity of each node is equal to the outgoing flow.

Attention: A Flow Network View



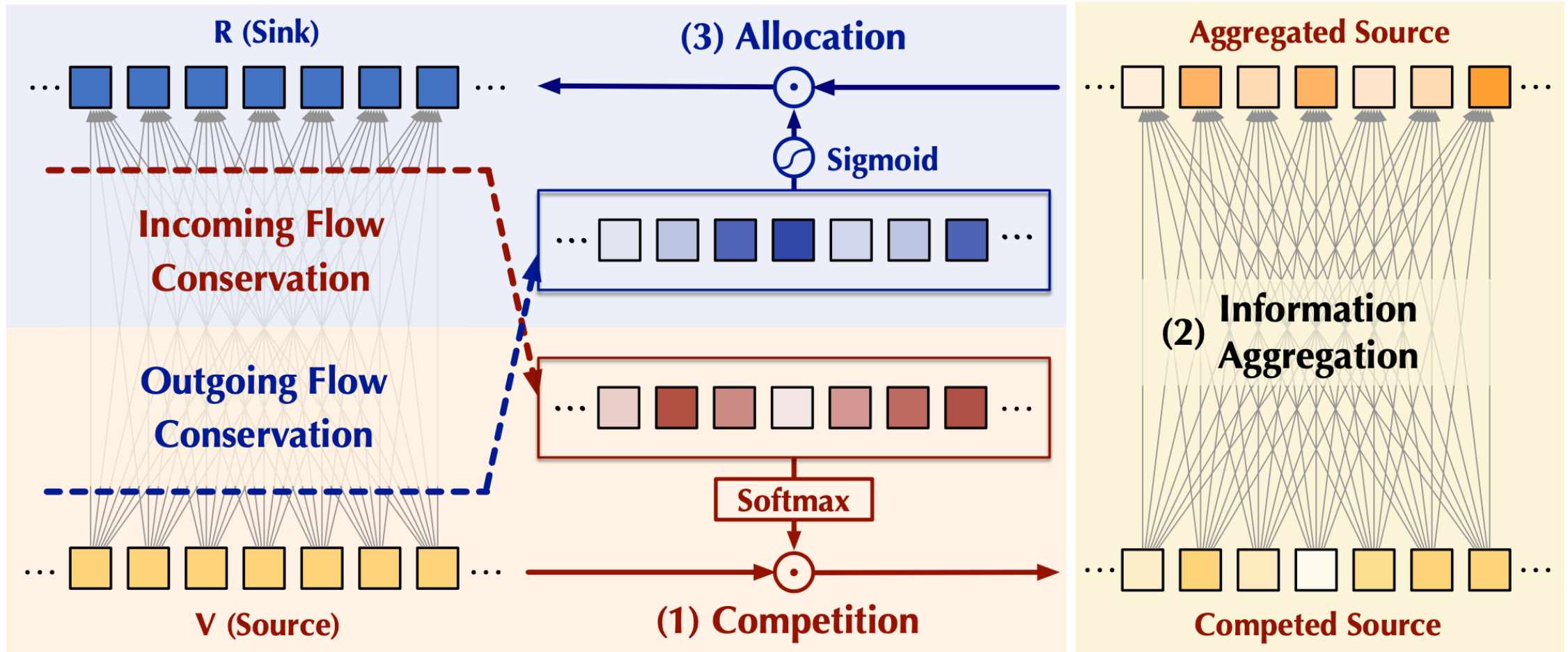
(a) Inner View

Attention: A Flow Network View



(b) Outer View

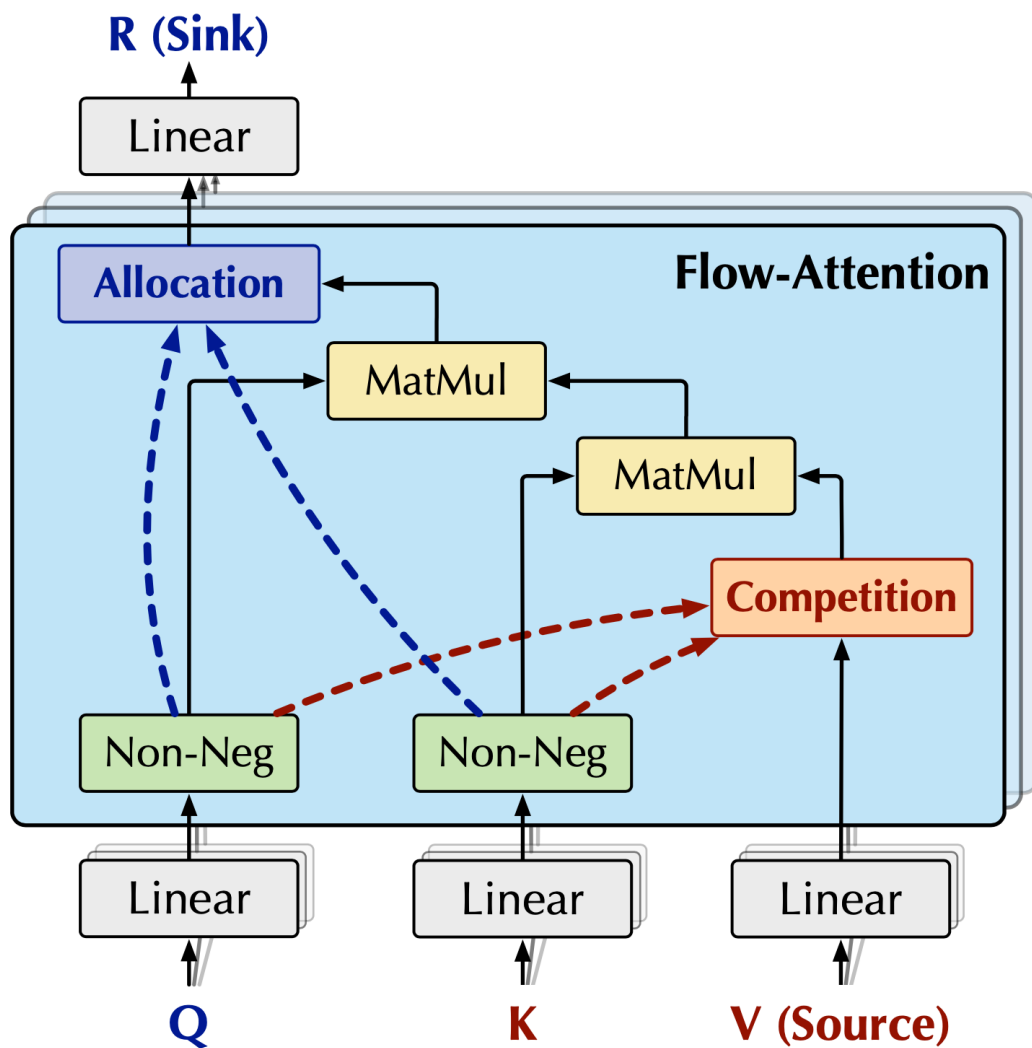
Conservation in Attention



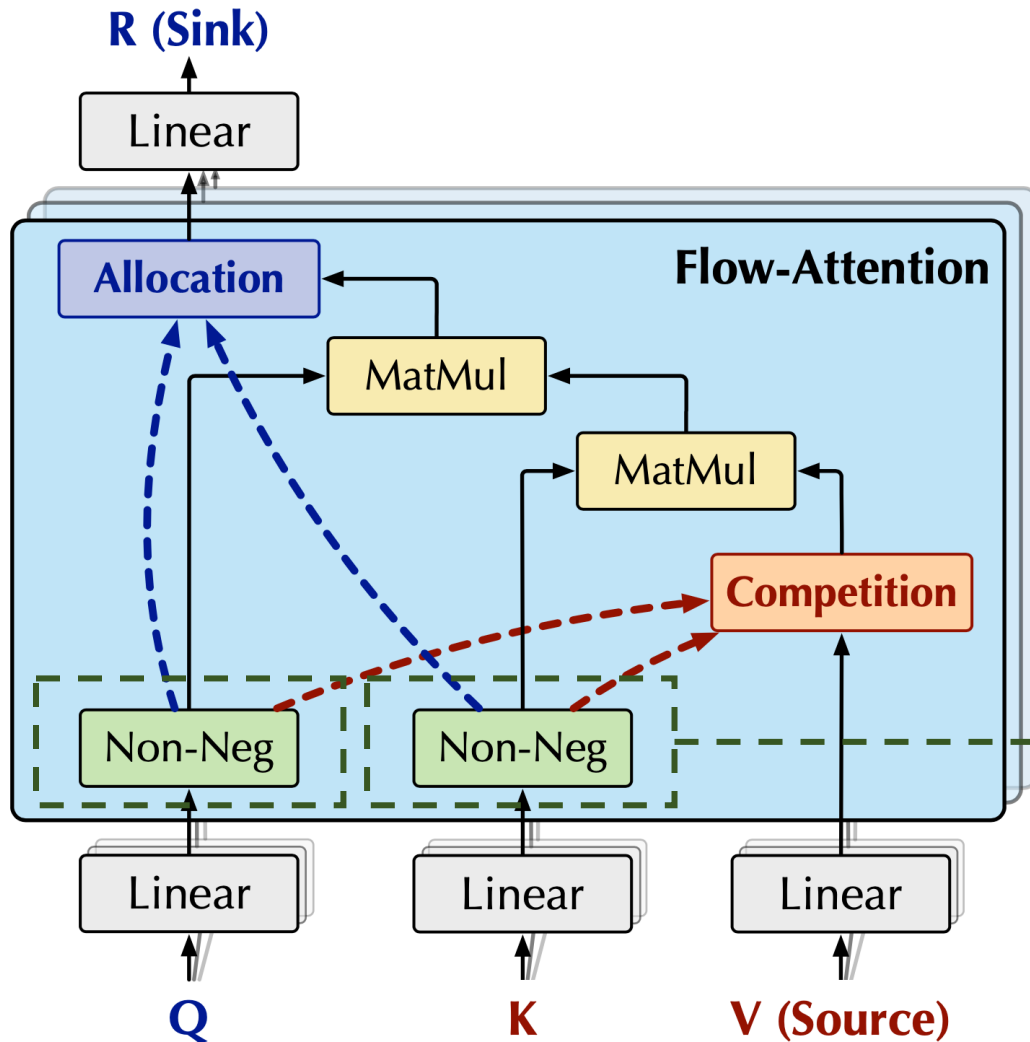
[Incoming Flow Conservation]: Competition among Source tokens

[Outgoing Flow Conservation]: Competition among Sink tokens

Flow-Attention

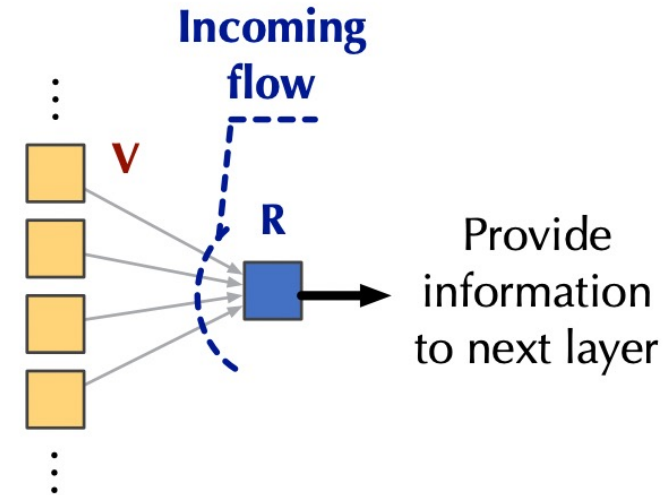
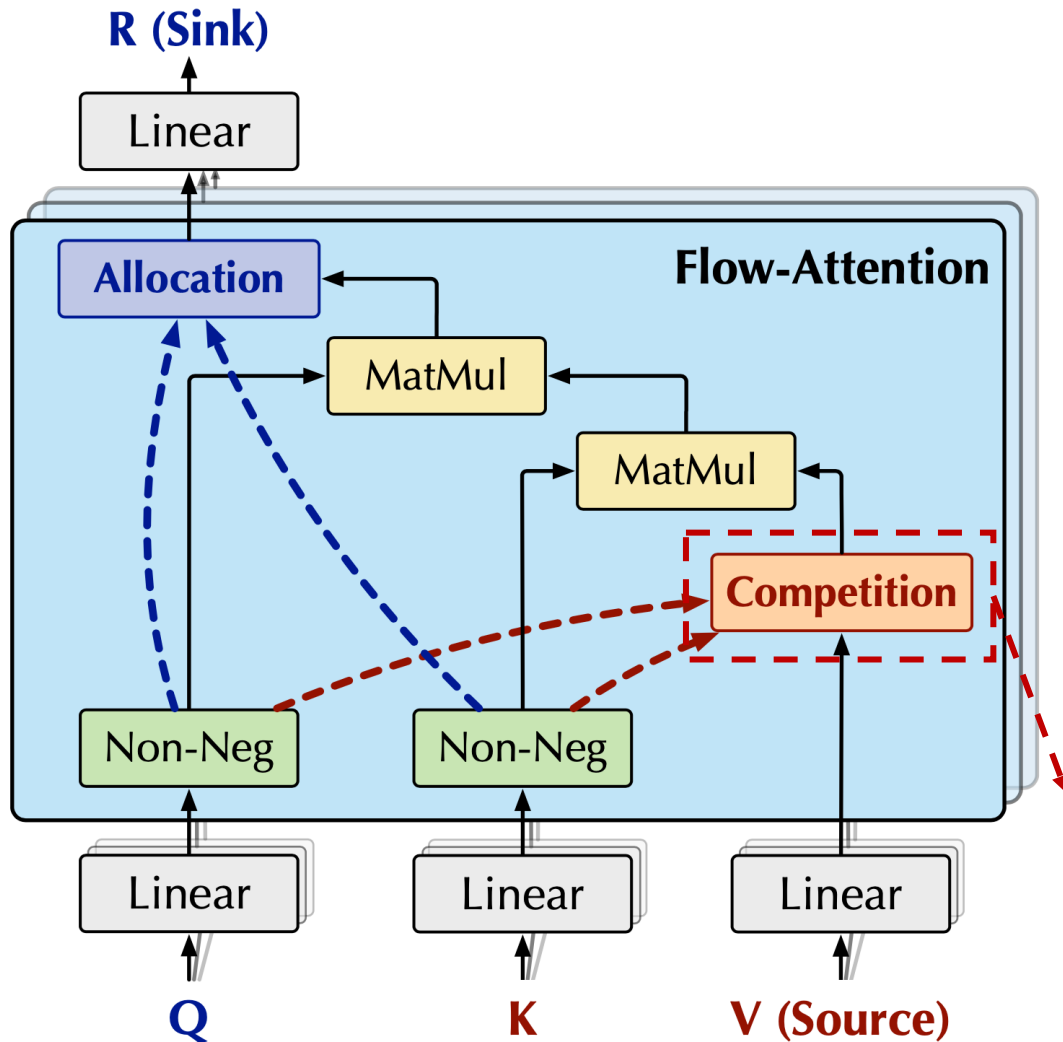


Flow-Attention



$$\phi(\cdot) = \text{Sigmoid}(\cdot) \text{ or } \phi(\cdot) = \text{ELU}(\cdot) + 1.0$$

Flow-Attention

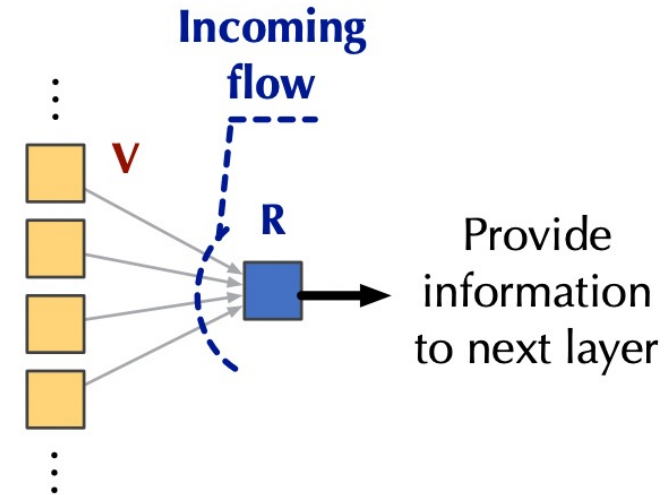
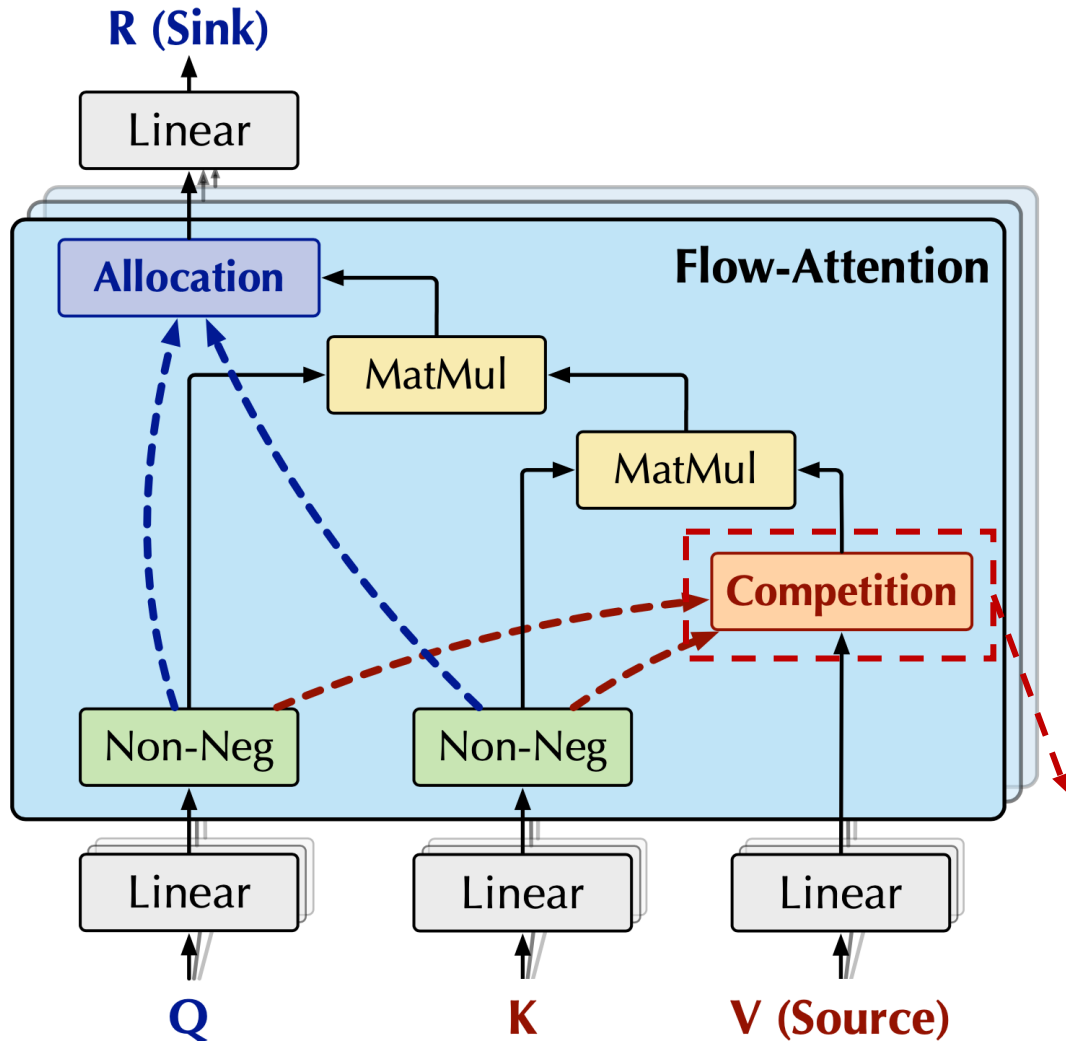


$$\text{Incoming flow: } I_i = \phi(Q_i) \sum_j \phi(K_j)^T$$

$$\text{Incoming flow conservation: } \frac{\phi(Q)}{I}$$

$$\text{Incoming flow: } \frac{\phi(Q_i)}{I_i} \sum_j \phi(K_j)^T = \frac{I_i}{I_i} = 1$$

Flow-Attention

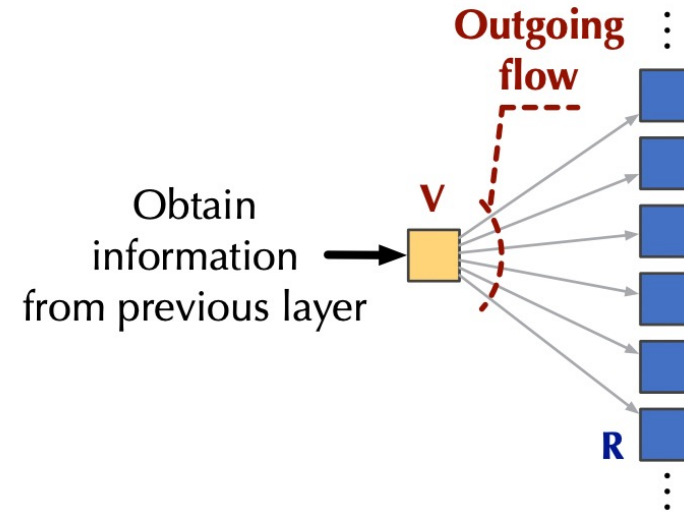
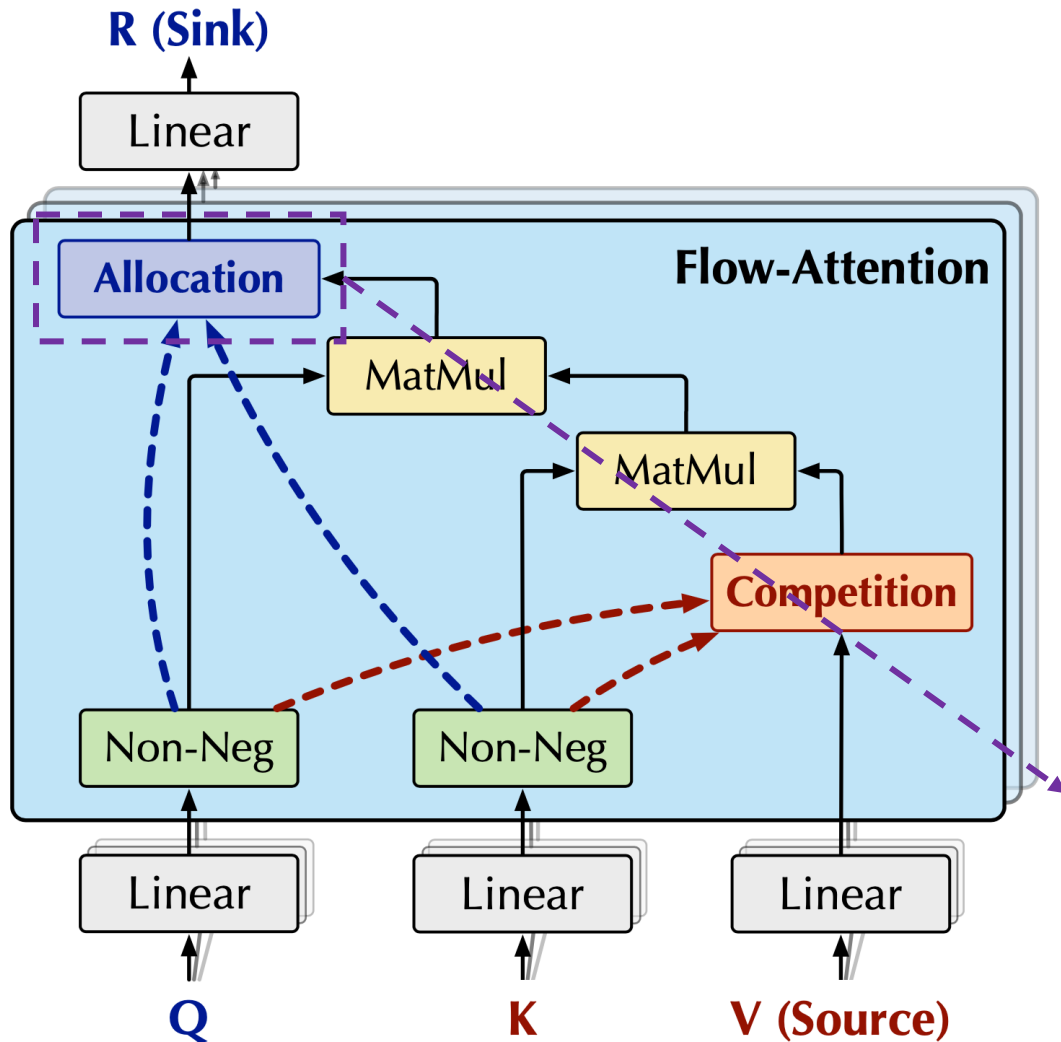


$$\text{Incoming flow: } I_i = \phi(Q_i) \sum_j \phi(K_j)^T$$

$$\text{Incoming flow conservation: } \frac{\phi(Q)}{I}$$

$$\text{Conserved outgoing flow: } \hat{o} = \phi(K) \sum_i \frac{\phi(Q_i)^T}{I_i}$$

Flow-Attention

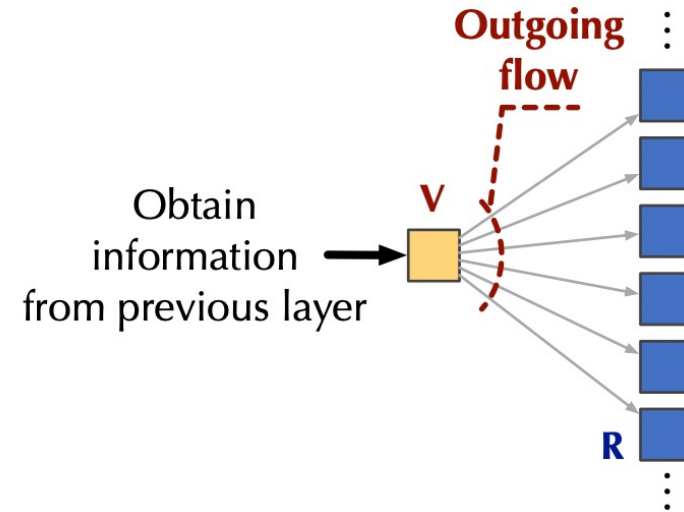
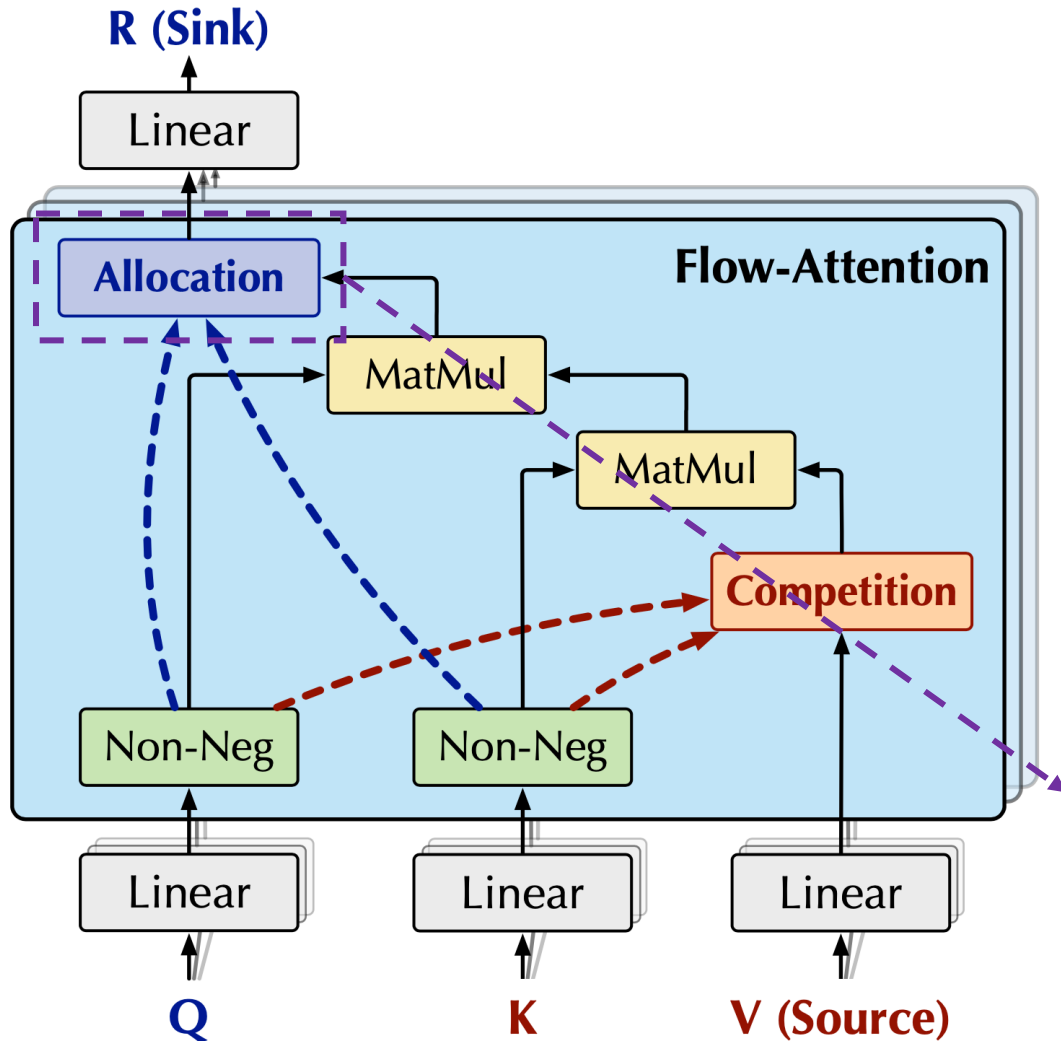


$$\text{Outgoing flow: } O_i = \phi(K_i) \sum_j \phi(Q_j)^T$$

$$\text{Outgoing flow conservation: } \frac{\phi(K)}{O}$$

$$\text{Outgoing flow: } \frac{\phi(K_i)}{O_i} \sum_j \phi(Q_j)^T = \frac{O_i}{O_i} = 1$$

Flow-Attention

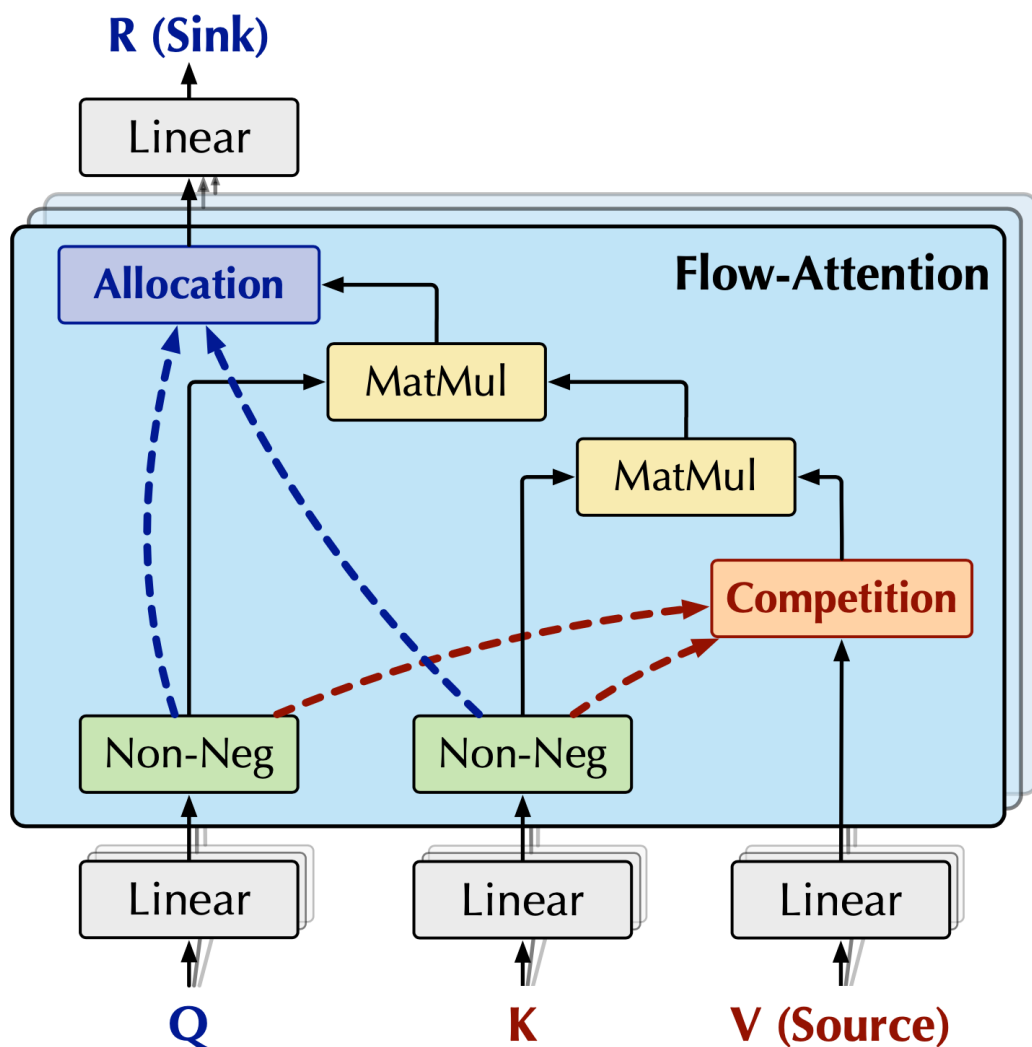


$$\text{Outgoing flow: } O_i = \phi(K_i) \sum_j \phi(Q_j)^T$$

$$\text{Outgoing flow conservation: } \frac{\phi(K)}{O}$$

$$\text{Conserved incoming flow: } \hat{I} = \phi(Q) \sum_j \frac{\phi(K_j)^T}{o_j}$$

Flow-Attention

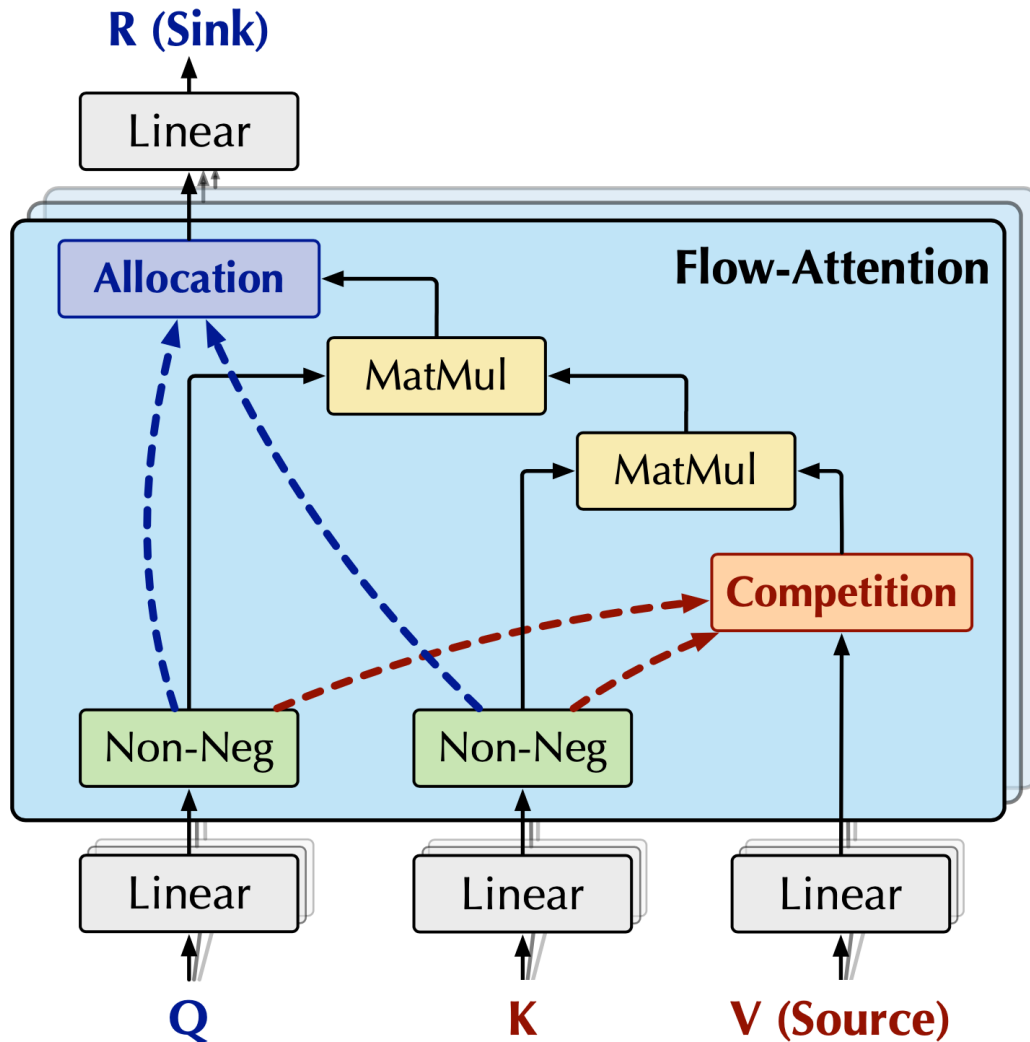


$$\text{Competition: } \hat{\mathbf{V}} = \text{Softmax}(\hat{\mathbf{O}}) \odot \mathbf{V}$$

$$\text{Aggregation: } \mathbf{A} = \frac{\phi(\mathbf{Q})}{\mathbf{I}} (\phi(\mathbf{K})^\top \hat{\mathbf{V}})$$

$$\text{Allocation: } \mathbf{R} = \text{Sigmoid}(\hat{\mathbf{I}}) \odot \mathbf{A},$$

Flow-Attention



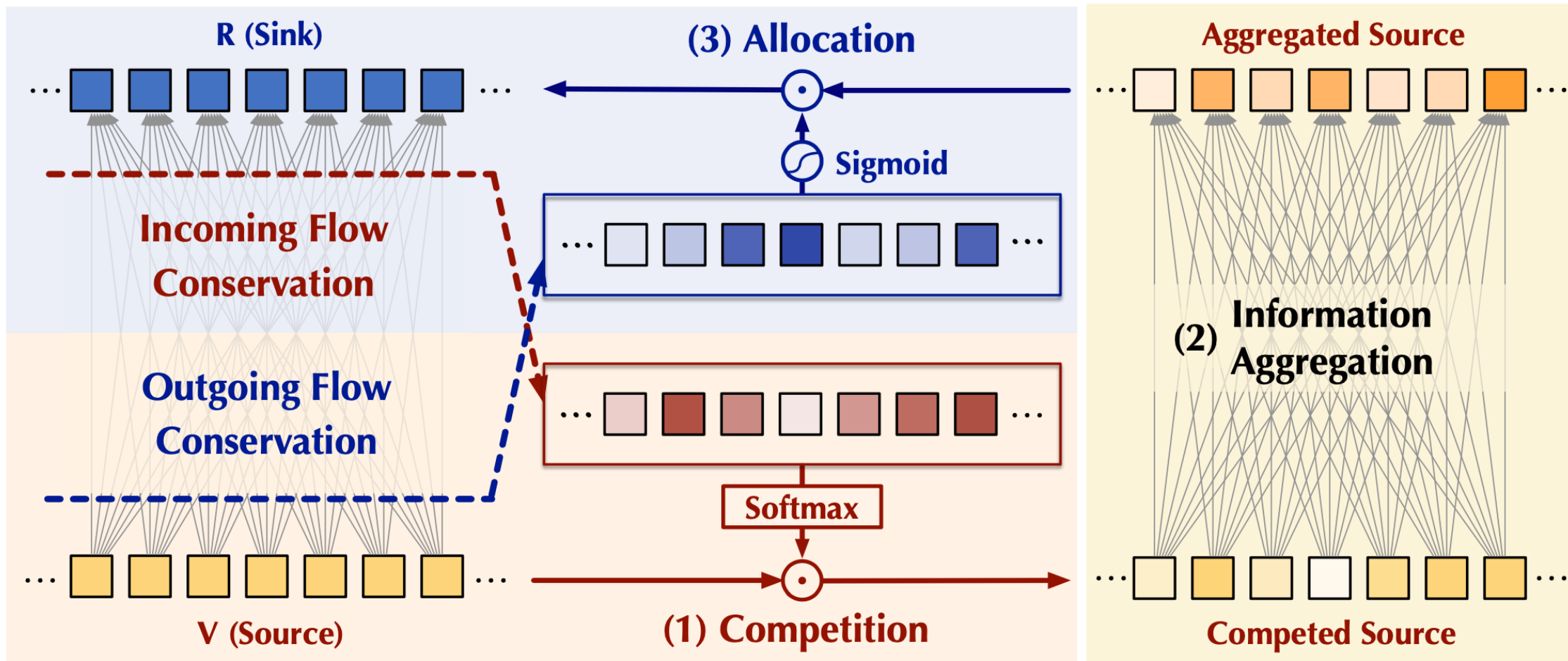
$$\text{Competition: } \hat{\mathbf{V}} = \text{Softmax}(\hat{\mathbf{O}}) \odot \mathbf{V}$$

$$\text{Aggregation: } \mathbf{A} = \frac{\phi(\mathbf{Q})}{\mathbf{I}} (\phi(\mathbf{K})^\top \hat{\mathbf{V}})$$

$$\text{Allocation: } \mathbf{R} = \text{Sigmoid}(\hat{\mathbf{I}}) \odot \mathbf{A},$$

Successfully bring the Competition Mechanism Into Attention design to avoid trivial attention

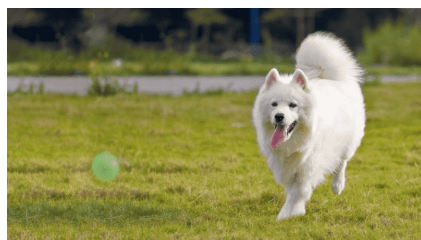
Efficiency and Universality



[Efficiency]: All the calculations are **in linear complexity**.

[Universality]: The whole design is based on flow network **without specific inductive biases**.

Flowformer Experiments



Image



Language



Time
Series



Agent
Trajectory

BENCHMARKS	TASK	VERSION	LENGTH
LRA (2020c)	SEQUENCE	NORMAL	1000~4000
WIKITEXT (2017)	LANGUAGE	CAUSAL	512
IMAGENET (2009)	VISION	NORMAL	49~3136
UEA (2018)	TIME SERIES	NORMAL	29~1751
D4RL (2020)	OFFLINE RL	CAUSAL	60

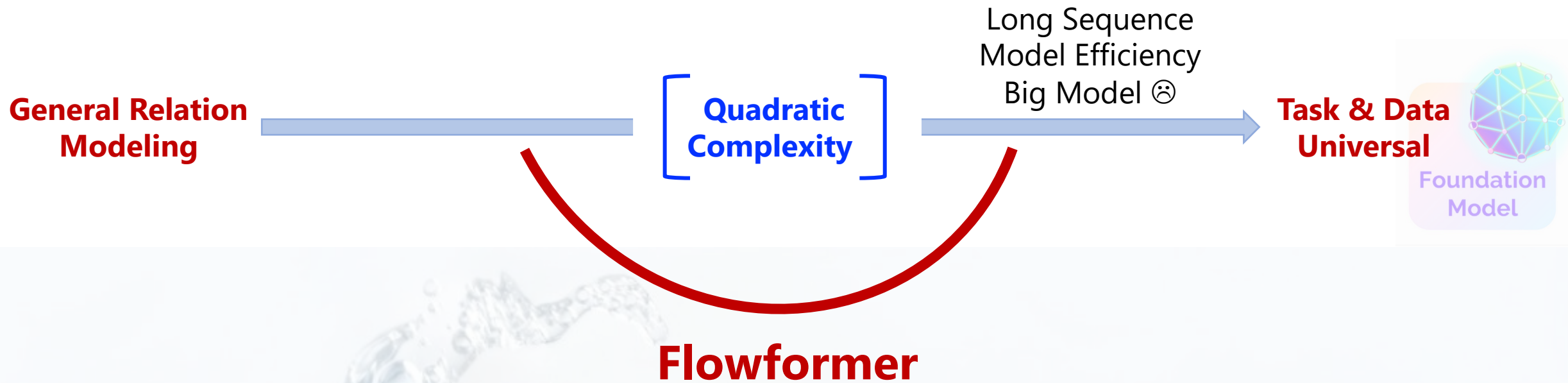
- Extensive tasks (covering 5 mainstream tasks)
- Normal and causal versions
- Various sequence lengths (29-4000)
- Extensive baselines (20+)

Flowformer Experiments

Task	Metrics	Flowformer	Performer	Reformer	Vanilla Transformer
Long Sequence Modeling (LRA)	Avg Acc (%) ↑	56.48	51.41	50.67	OOM
Vision Recognition (ImageNet-1K)	Top-1 Acc (%) ↑	80.6	78.1	79.6	78.7
Language Modeling (WikiText-103)	Perplexity ↓	30.8	37.5	33.6	33.0
Time series classification (UEA)	Avg Acc (%) ↑	73.0	71.5	71.9	71.9
Offline RL (D4RL)	Avg Reward ↑ Avg Deviation ↓	73.5 ± 2.9	63.8 ± 7.6	63.9 ± 2.9	72.2 ± 2.6

Strong performance on all five mainstream tasks within the linear complexity.

Summary



Linear complexity w.r.t. sequence length

Based on flow network & **without specific inductive biases**

Strong performance in **Long Sequence, CV, NLP, Time Series, RL**

Open Source

The screenshot shows the GitHub repository page for `thuml/Flowformer`. The repository is public and has 6 watches, 3 forks, and 41 stars. The main branch is `main` with 1 branch and 0 tags. The repository contains a table of files and folders, a README.md file, and a list of contributors. The README.md file is titled "Flowformer (ICML 2022)" and describes the project as "Flowformer: Linearizing Transformers with Conservation Flows". It mentions that Transformers have achieved impressive success in various areas, but the attention mechanism has a quadratic complexity, which is significantly impeding Transformers from dealing with numerous tokens and scaling up to bigger models. The project proposes Flowformer, which is described as "Linear complexity", "Without specific inductive bias", and "Task-universal".

File/Folder	Commit Message	Commit Date
Flowformer_CV	Update README.md	6 days ago
Flowformer_LRA	Update README.md	10 days ago
Flowformer_RL	Update trajectory_gpt2.py	4 days ago
Flowformer_TimeSeries	Update README.md	10 days ago
pic	update RL readme	6 days ago
.gitignore	Initial commit	16 days ago
Flow_Attention.py	Update Flow_Attention.py	5 days ago
LICENSE	Initial commit	16 days ago
README.md	Update README.md	2 days ago

Flowformer (ICML 2022)

Flowformer: Linearizing Transformers with Conservation Flows

Transformers have achieved impressive success in various areas. However, the attention mechanism has a quadratic complexity, significantly impeding Transformers from dealing with numerous tokens and scaling up to bigger models. In pursuing the **linear complexity** and **task-universal** foundation model, we propose Flowformer [paper] with the following merits:

- **Linear complexity** w.r.t sequence length, can handle extremely long sequence (over 4k tokens)
- **Without specific inductive bias**, purely derived from the flow network theory
- **Task-universal**, showing strong performance in **Long sequence, Vision, NLP, Time series, RL**.

Contributors 2

- wuhaixu2016
- Manchery Jialong Wu

Languages

- Python 98.7%
- Shell 1.3%

<https://github.com/thuml/Flowformer>

Complete benchmarks & datasets & scripts

Thank You!

whx20@mails.tsinghua.edu.cn



长按关注，获取最新资讯