MODEL SHAPLEY: Find Your Ideal Parameter Player via One Gradient Backpropagation

Xu Chu[1,3,4] chu_xu@pku.edu.cn
Xinke Jiang[1,2,3] xinkejiang@stu.pku.edu.cn
Rihong Qiu[1,2,3] rihongqiu@stu.pku.edu.cn
Jiaran Gao[1,2] jiarangao@stu.pku.edu.cn
Junfeng Zhao[1,3,5] zhaojf@pku.edu.cn

NEURAL INFORMATION PROCESSING SYSTEMS

北京大學 PEKING UNIVERSITY

The Scale of Modern AI & The 「Parameter Equality」 Myth

- Modern Deep Neural Networks (DNNs), especially Large Language Models (LLMs), boast hundreds of millions to billions of parameters.
- This sheer scale presents challenges in:
  - Understanding
  - Optimizing
  - Deploying
- **Fundamental Observation:** "Not all parameters are created equal." Their contributions to model performance vary significantly.

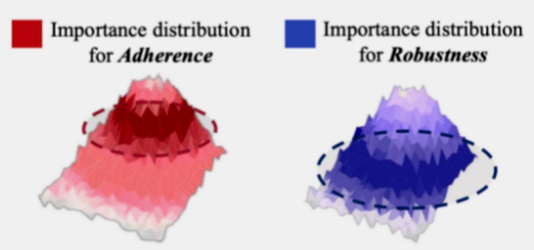
Motivation: Parallel Distributed Processing (PDP) & Synergy

The PDP Paradigm

The emergent behavior of a network arises from the **collective interactions** among numerous parameters.

Neurons, Attention Heads and Layers...

- Individually, a parameter might appear unimportant.
- Yet, it can become **crucial** when acting in synergy with others.
- This necessitates methods that can precisely identify which parameters, or groups thereof, drive performance.



A case in RAG--some parameters are important for adherence and some are for roubstness ability

MODEL SHAPLEY

A novel approach to quantify parameter importance by recasting the problem within cooperative game theory.

- Each model parameter is treated as a "player" in a game.
- Importance is measured by its **Shapley value**: the average marginal improvement it contributes across all possible subsets (coalitions) of other parameters.
- This inherently accounts for both individual contributions and crucial synergistic interactions, aligning with the PDP perspective.
- The Shapley value is a principled metric satisfying desirable fairness axioms, ensuring robust and equitable attribution.

Definition

The Shapley value $\phi_\theta(U)$ for a parameter θ quantifies its **average marginal contribution** to the utility function, computed over all possible subsets of other parameters that do not include θ.

$$\phi_\theta(U) := \sum_{\Theta_S \subseteq \Theta \setminus \{\theta\}} \frac{|\Theta_S|!(M - |\Theta_S| - 1)!}{M!} [U(\Theta_S \cup \{\theta\}) - U(\Theta_S)]$$

The Challenge: Combinatorial Explosion

- Despite its theoretical appeal, exact Shapley computation is a nightmare for DNNs.
- Evaluating $U(\Theta_S \cup \{\theta\}) - U(\Theta_S)$ for **all** 2^{M-1} subsets Θ_S is required for each parameter.
- For LLMs with M in millions or billions (0.5B, 7B, 670B parameters!), this is $O(2^M)$ complexity – utterly infeasible.
- Each evaluation might mean:
  - Removing/zeroing parameters.
  - Re-evaluating or even re-training the model.
- This has historically limited Shapley values for parameter-level attribution, motivating scalable approximations like MODEL SHAPLEY.

Core Idea: Path-Integrated Approximation

How to avoid exponential evaluations?

- Instead of discrete parameter removal and retraining, MODEL SHAPLEY estimates the change in training loss using a **path-integrated formulation**.
- Parameter removal is modeled as a continuous trajectory in parameter space.
- The total effect is computed by integrating the gradient along this path.
- **Theorem 4.1 (Path-Integrated Loss from paper):** For a perturbed configuration Θ' along a linear path from Θ^τ: $\mathcal{L}(\Theta'; x, y) = \mathcal{L}(\Theta^\tau; x, y) + \int_0^1 \nabla_\Theta \mathcal{L}(\Theta_t; x, y)^\top \frac{d\Theta_t}{dt} dt$, where $\Theta_t := \Theta^\tau + t(\Theta' - \Theta^\tau)$.

Single Parameter Removal (Theorem 4.2 from paper)

Loss change $\Delta\mathcal{L}(\theta_i^\tau)$ from removing θ_i^τ: $\Delta\mathcal{L}(\theta_i^\tau) \approx -g_i^\tau \theta_i^\tau + \frac{1}{2} w_{ii}^{(i)} H_{ii}^\tau (\theta_i^\tau)^2$ (g_i^τ: gradient, H_{ii}^τ: Hessian diagonal, $w_{ii}^{(i)}$: curvature path weight)

Parameter Subset Removal (Theorem 4.4 from paper)

Loss change $\Delta\mathcal{L}(\Theta^S)$ from removing subset $\mathcal{S}$: $\Delta\mathcal{L}(\Theta^S) \approx -\sum_{i \in \mathcal{S}} g_i^\tau \theta_i^\tau + \frac{1}{2} \sum_{i,j \in \mathcal{S}} w_{ij}^{(S)} H_{ij}^\tau \theta_i^\tau \theta_j^\tau$ (Now includes off-diagonal Hessian H_{ij}^τ for interactions)

Aggregating marginal utilities (derived from loss changes) over all subsets yields:

MODEL SHAPLEY (Equation 7)

$$\phi_i = \underbrace{-\mathbf{g}_i^\tau \theta_i}_{\text{(1) Individual Importance}} \underbrace{-\frac{1}{2}\mathbf{w}_{ii}^{(i)} \theta_i^2 \mathbf{H}_{ii}^\tau - \frac{1}{2}\theta_i \sum_{j \neq i} \mathbf{w}_{ij}^{(S)} \mathbf{H}_{ij}^\tau \theta_j}_{\text{(2) Cooperative Interactions}}$$

Computational Efficiency (Remark 4.9 from paper)

Estimating all $\{\phi_i\}$ requires only:

- 1 Forward Pass (loss, parameter values)
- 1 Backward Pass (gradients)
- 1 Hessian extraction (or approximation, e.g., HVP)

This makes Shapley-style attribution tractable for large models!

Algorithm 2 Parameter-wise Shapley Value Estimation with Gradient Similarity

Require: Model parameters $\Theta^\tau = \{\theta_1^\tau, \ldots, \theta_M^\tau\}$, loss function $\mathcal{L}$, mini-batch size B, smoothing coefficient α, total steps T
Ensure: EWMA-based Shapley value estimates $\{\widehat{\phi}_i^T\}_{i=1}^M$
1: Initialize $\widehat{\phi}_i^0 \leftarrow 0$, for all $i = 1, \ldots, M$
2: **for** $\tau = 1$ to T **do**
3: Sample mini-batch $\mathcal{B}^\tau = \{(x_j, y_j)\}_{j=1}^B$
4: Compute gradient: $\mathbf{g}^\tau \leftarrow \nabla_\Theta \mathcal{L}_B(\Theta^\tau)$
5: Compute curvature approximation: $\mathbf{H}^\tau \leftarrow \mathbf{g}^\tau \times \mathbf{g}^\tau$
6: $\phi^{(1)} \leftarrow -g^\tau \cdot \theta^\tau$
7: $\phi^{(2)} \leftarrow -\frac{1}{2}\theta^\tau \cdot (H^\tau \times \theta^\tau)$
8: $\phi^\tau \leftarrow \phi^{(1)} + \phi^{(2)}$
9: $\widehat{\phi}^\tau \leftarrow (1-\alpha)\widehat{\phi}^{\tau-1} + \alpha \cdot \phi^\tau$
10: **end for**
11: **return** $\{\widehat{\phi}_i^T\}_{i=1}^M$

Experiments

Table 1: Evaluation of different inference and training methods across models and datasets.

Method	VIT-Base/16 (CV)		Qwen2.5-3B (NLP)		Qwen2.5-7B (NLP)	
	CIFAR-100	ImageNet	GSM8K	MMLU	GSM8K	MMLU
Pretrain	79.69	76.14	45.57	60.81	72.48	73.06
Inference (Deactivate Neurons)						
Random	08.39	18.25	04.47	50.11	19.94	66.15
Gradient	77.82	65.99	37.53	50.31	46.70	66.86
Gradient Trace	76.65	67.62	36.09	51.99	72.71	68.01
MODEL SHAPLEY	**80.84**	**70.33**	**38.06**	**52.08**	**73.39**	**68.93**
Full Fine-Tune	85.31	78.09	54.89	63.08	72.55	73.56
Training (Freeze Neurons)						
Random	84.27	79.76	46.98	60.68	60.80	68.44
Gradient	84.64	79.63	47.57	61.35	61.87	69.08
Gradient Trace	84.69	79.57	47.08	63.59	61.41	70.79
MODEL SHAPLEY	**86.53**	**79.82**	**47.76**	**63.72**	**62.02**	**73.89**

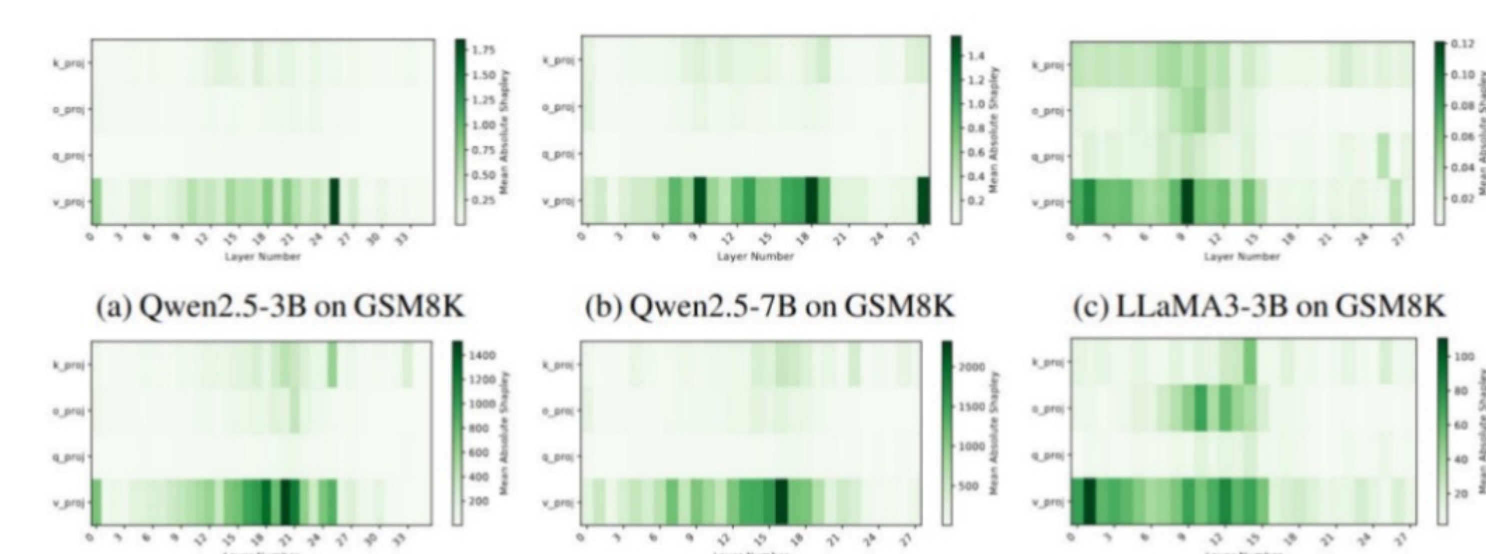
Table 2: Evaluation of different inference and training methods across models and datasets on GSM8K.

Quantization	INT4 (W4A16)		INT8 (W8A8)		FP8 (WA-FP8)	
	Runtime(min)	Accuracy	Runtime(min)	Accuracy	Runtime(min)	Accuracy
Qwen 2.5-Instruct (7B) No Compression Accuracy: 72.48						
GPTQ	71.73	62.70	**53.86**	70.58	**52.19**	74.15
OBD	**70.98**	62.55	57.88	71.27	53.88	71.42
MODEL SHAPLEY	74.67	**63.23**	60.20	**72.93**	58.31	**74.83**
Qwen 2.5-Instruct (14B) No Compression Accuracy: 77.41						
GPTQ	132.28	**65.20**	**71.68**	75.66	78.63	76.95
OBD	128.99	62.77	94.90	75.36	**73.74**	77.48
MODEL SHAPLEY	**98.84**	63.84	76.46	**76.27**	84.75	**78.85**

Model Shapley demonstrates exceptional performance in two core fields: Computer Vision (CV) and Natural Language Processing (NLP).

It plays a steady and critical rolein the model training phase, and in the inference and even model quantization phase.

- **Figure 1 (Layer-wise Shapley):** Reveals task-specific importance patterns in q/k/v/o projections and layers for different models (Qwen, LLaMA).



(a) Qwen2.5-3B on GSM8K (b) Qwen2.5-7B on GSM8K (c) LLaMA3-3B on GSM8K
(d) Qwen2.5-3B on MMLU (e) Qwen2.5-7B on MMLU (f) LLaMA3-3B on MMLU

Figure 1: Layer-wise shapley value for q/k/v/o projection.