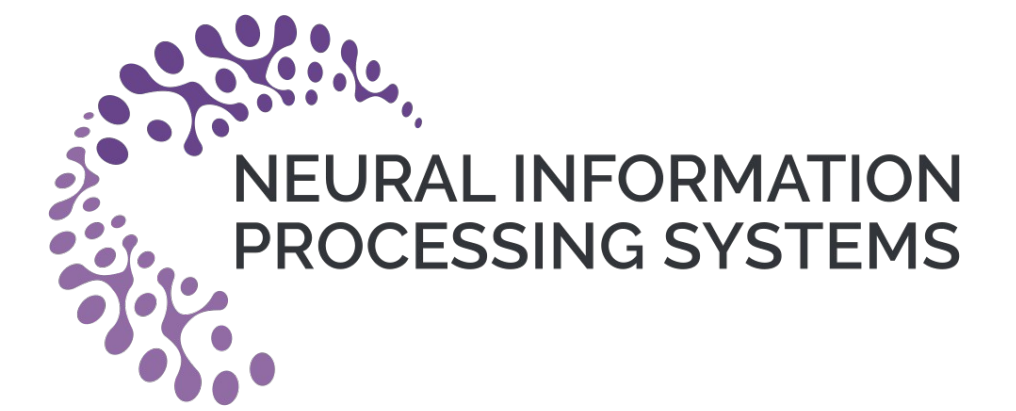




HALO: Homotopy-Augmented Layer Optimization for Stable LLM Supervised Post-training

Mengxiang Zhang and Lingyuan Liu



Motivation

Full-parameter SFT and KD adapt LLMs but risk high compute, instability, and overfitting.

Discrete unfreezing cuts updates, but abrupt activation causes optimization shock and representation bottlenecks.

Contributions

HALO couples backward output-to-input unfreezing with stage-conditioned logit continuation, standalone or with PEFT.

Conditional theory proves fixed- t stationarity and bounds finite-stage tracking via step size and budget.

Across five model families and three domains, HALO improves SFT/KD generalization and transition stability.

Theoretical Insight

Under cross-entropy, bounded logit gradients, and diminishing steps, fixed- t weighted-average expected squared gradient norms vanish.

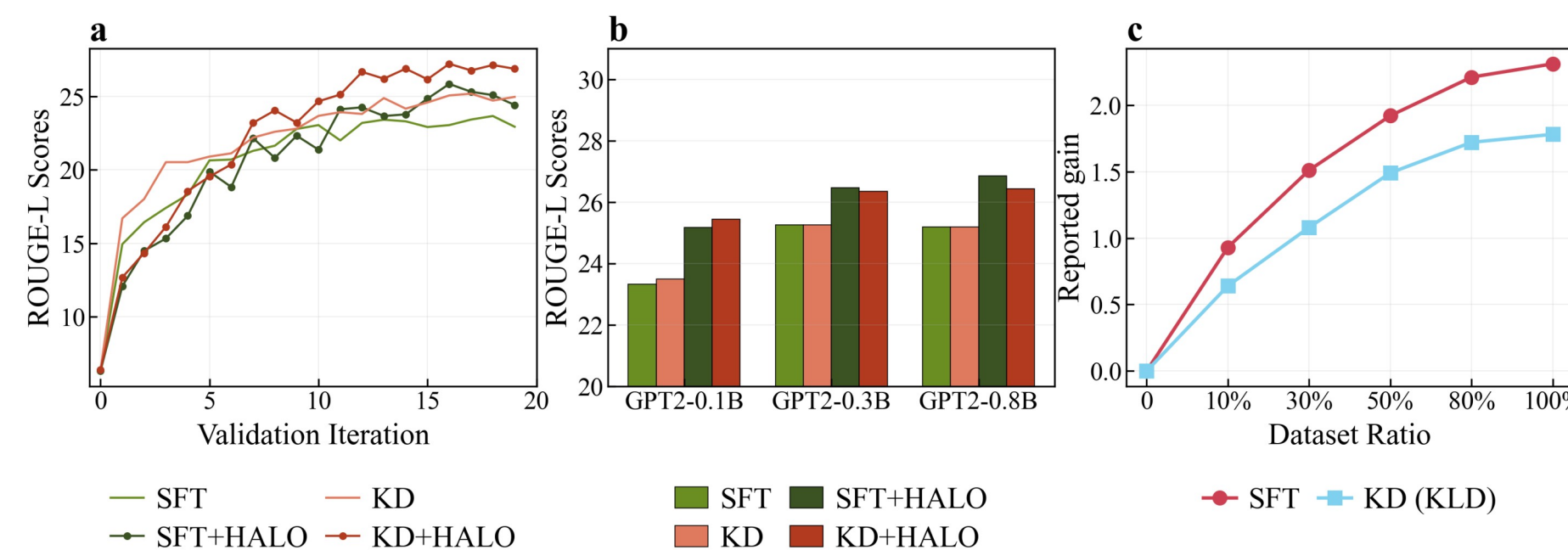
Under local PL and L-smoothness assumptions, finite-stage tracking balances gradient noise against path lag.

$$\delta \lesssim \frac{LG^2}{4\mu} \eta + \frac{C}{2\mu\eta N_s}$$

Efficiency

On GPT2-0.1B (dolly-15k, four A100s), HALO cuts block-epoch updates 41.7% and runtime 24.1%–29.6% while improving SFT/KD validation-iteration convergence.

Across GPT2-0.1B, 0.3B, and 0.8B capacities and 10%–100% dolly-15k fractions, HALO delivers consistent SFT/KD ROUGE-L improvements and reported gains.



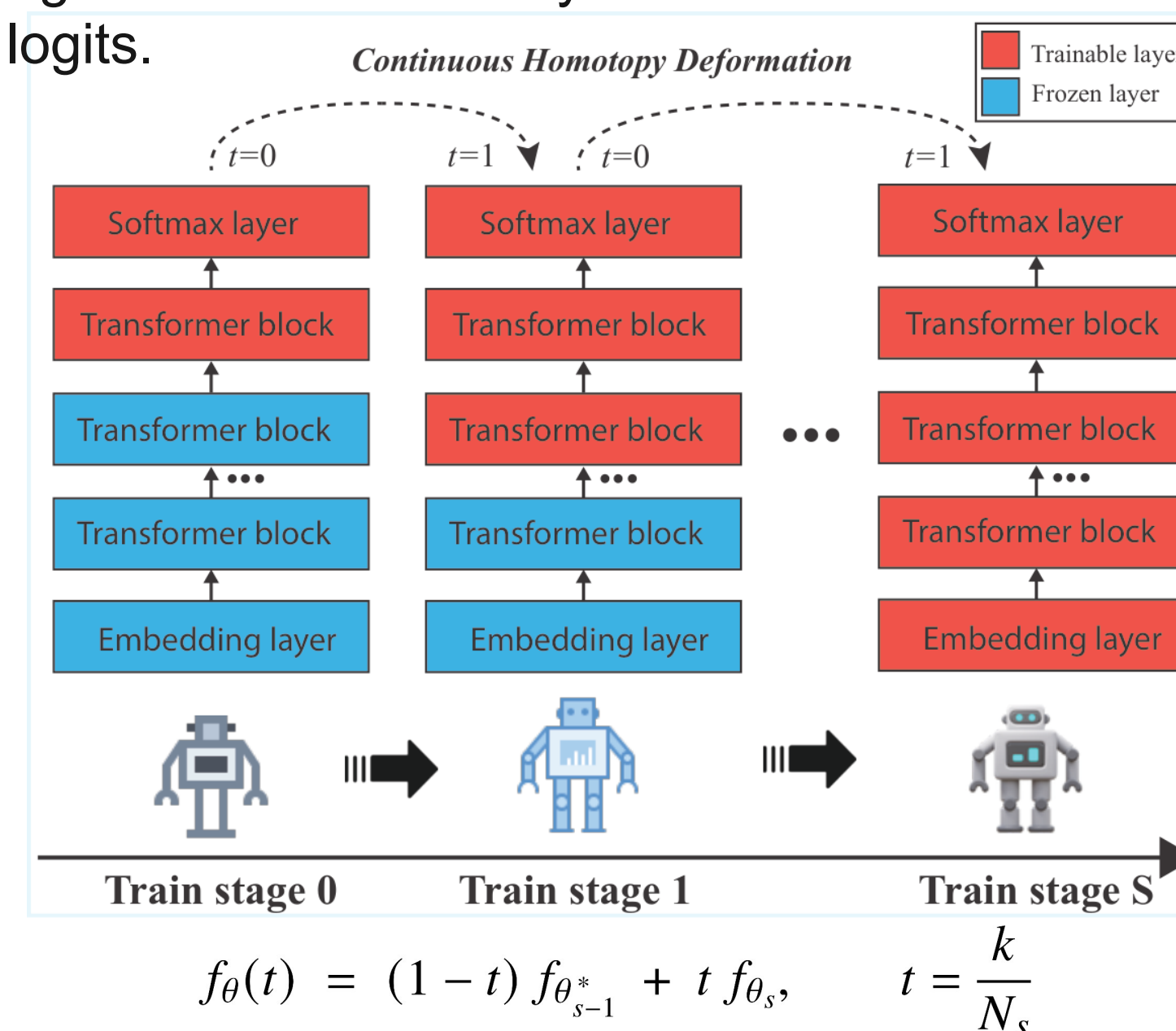
When applied over LoRA/AdaLoRA, HALO shares the frozen base model and snapshots only lightweight adapter weights.

HALO Framework

HALO expands trainable layers backward from output head to input embeddings.

Each stage interpolates logits between the frozen previous-stage anchor and active network.

Within-stage t increases linearly while SFT or KD loss evaluates mixed logits.



Evaluating loss on interpolated logits aligns boundary gradients with the anchor residual, adjusting both direction and magnitude.

Stability & Ablation

On GPT2-0.1B, HALO cuts update spikes and gradient CV while raising effective rank.

On GPT2-0.1B, linear logit continuation beats FFT, discrete unfreezing, and matched smooth controls.

Method	Spike $R\Delta\theta$ ↓	Grad. CVg ↓	Rank reff ↑
Discrete unfreezing	4.3	1.74	27.9
+ New-group warmup	3.8	1.52	33.1
HALO	1.5	0.35	45.3

Method	SFT Avg. R-L ↑	KD Avg. R-L ↑
Baseline (FFT)	16.80	18.10
HALO (Standard, Linear t)	18.32	19.09
HALO (No Homotopy / Discrete)	15.32	17.21
Discrete + New-Group LR Warmup	17.74	18.63
Discrete + Parameter Interp.	17.57	18.75

Main Results

Standalone and plug-in HALO improve Qwen2.5-0.5B SFT/KD instruction following.

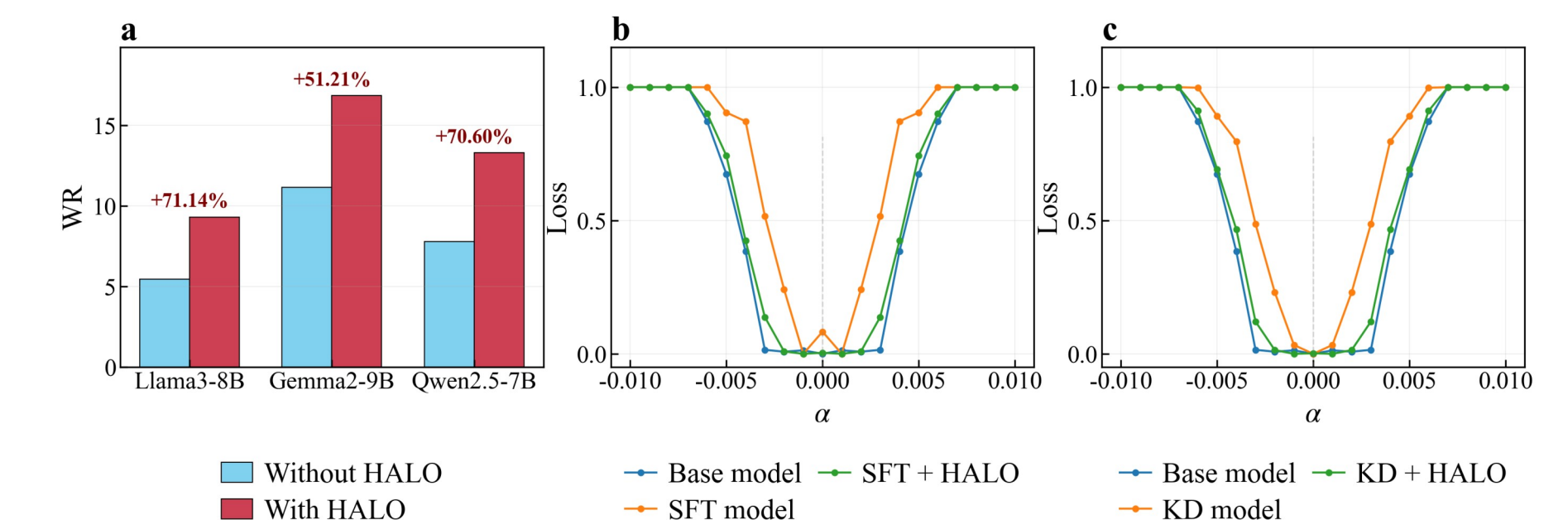
Method	SFT R-L ↑	SFT WR ↑	KD R-L ↑	KD WR ↑
FFT	20.79	45.17	24.59	49.80
ULMFIT	20.92	45.12	24.64	50.36
IR-Tuning	21.23	44.92	24.84	50.04
IR-Tuning+HALO	22.05	46.36	25.58	51.66
AdaLoRA	21.01	45.72	24.71	50.23
AdaLoRA+HALO	21.50	47.77	25.44	51.59
HALO	22.61	48.70	25.35	51.83

LoRA-based HALO raises SFT and KD pass@1 on math and code.

Method	GSM8K	MATH	HumanEval	MBPP
SFT	76.2	53.4	43.1	60.7
SFT + HALO	77.6	55.5	45.5	61.8
KLD	77.9	55.2	45.1	61.3
KLD + HALO	78.9	57.3	46.4	62.1

Generalization

HALO lifts AlpacaEval v2 win rates across Llama3-8B, Gemma2-9B, and Qwen2.5-7B while broadening GPT2-0.1B SFT and KD loss basins.



LoRA SFT with HALO boosts 7B–9B win rates by 51%–71%, while broader GPT2-0.1B loss basins accompany improved retention–adaptation trade-offs.

Take-home Message

Logit continuation stabilizes backward unfreezing, cutting SFT/KD updates and improving generalization given sufficient stage steps.