

Peer Review

Review of: "LoLDU: Low-Rank Adaptation via Lower-Diag-Upper Decomposition for Parameter-Efficient Fine-Tuning"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The authors propose LoLDU, a parameter-efficient fine-tuning method that factorises each frozen weight matrix as $W = PL\text{diag}(z)U$ (Section III.A), takes the top- r slice of that factorisation as a trainable "Decomposed Subspace Matrix" (DSM), keeps the remainder as a frozen "Residual Subspace Matrix" (RSM, Eq. 7), and then trains only the r diagonal entries of $\text{diag}(z)$ plus a scalar σ . Section III.E puts the trainable storage at $O(r+1)$ per matrix, against $O(mr+rn)$ for LoRA. The pitch is that initialising from the weight itself, instead of from random matrices, gives faster convergence, and that the triangular factors carry an orthogonality property that preserves pre-trained knowledge.

The evaluation spans four settings: RoBERTa-Base on GLUE (Table I), LLaMA2-7B trained on Alpaca and scored with INSTRUCTEVAL (Table III), ViT-Base on six image classification sets (Tables II, IV, V, XI, XII) and Stable Diffusion v1.5 personalisation on seven concepts (Table VI, Figures 5 and 6). The headline claims are "2600 times" fewer trainable parameters than regular PEFT and "0.00025% of parameters without any performance degradation" (Abstract, Section I, Section V). I like the idea. Training only the pivots of an LU factorisation is a cheap, simple and genuinely different parameterisation from the SVD family, and the authors released code.

A note on what I could read. On Qeios, the results tables (Tables I to VI, XI and XII) are embedded as images, and as rendered in the text version I worked from, they are not legible; only their captions survive. The hyperparameter tables (VII to X) are legible. Wherever I quote a result below, it comes from the prose of Sections I and IV or from the Figure 1 caption, and I say so.

2. Whether the conclusion holds on the data

Several things here are good. The method spans four model families and two modalities, which is more than most PEFT papers attempt. The ablations cover initialisation (Table V), the scaling factor, rank (Table XII, Figure 4) and learning rate (Table XI, Figure 7), and the rank sweep in the Figure 1 caption goes all the way down to $r = 1$, where the model has 24 trainable parameters. I checked that figure: ViT-Base has 12 blocks, the method is applied to query and value (Table IX), so $12 \times 2 \times 1 = 24$. It matches. The observation that accuracy saturates beyond rank 256 (Section IV.E, rank paragraph) is a useful practical finding.

Where I think the text claims more than it shows:

The orthogonality argument. Section I, Section III.A and Section III.C state that the LDU decomposition "preserves the orthogonality of the lower and upper triangular matrices." This is not true as a matter of linear algebra. A unit lower triangular matrix is orthogonal only if it is the identity; a general L or U from Gaussian elimination is not orthogonal, and its columns are not mutually orthogonal either. Only P is orthogonal. The link to OFT [21] and to hyperspherical energy therefore does not transfer. I would remove the orthogonality claim and motivate the method by what it does have: a deterministic, weight-derived initialisation and a very small trainable set. The results can stand on their own.

Which rank the headline results use. Table VII (GLUE) and Table IX (ViT) both set Rank = 768, which equals the hidden size of RoBERTa-Base and ViT-Base. At $r = k = \min(m, n)$ the top- r slice is the whole factorisation, so the RSM of Eq. 7 is zero at initialisation and the method becomes a full-rank diagonal rescaling inside a fixed LU basis. That is a legitimate method. It is not a low-rank adaptation, though, and the title and Section III.B frame it as one. I would state this plainly and report GLUE and ViT results at, say, $r = 256$ as well, since Section IV.E already recommends "approximately one-third of the full rank."

Parameter percentages do not reconcile. I tried to reproduce them from the hyperparameter tables.

- Figure 1 caption says LoLDU $r = 768$ has "96.837% fewer trainable parameters" than LoRA $r = 16$. Section IV.C says LoLDU uses "3.173% of LoRA's parameters." These two cannot both hold: $100 - 3.173 = 96.827$, not 96.837. My own count from Table IX gives 18,432 against 589,824 (see the next point), or 3.125%, so neither figure matches exactly.
- Section IV.E(d) says Table II reports LoLDU at "0.21% of the parameters" and LoRA at "6.77%." With Table IX (rank 768, query and value) and 12 blocks, LoLDU trains $12 \times 2 \times 768 = 18,432$ parameters, and $18,432 / 85.8\text{M}$ (the full model size given in the Figure 1 caption) = 0.0215%. LoRA $r = 16$ on the same

two matrices trains $12 \times 2 \times 16 \times (768 + 768) = 589,824$, i.e. 0.687% of 85.8M. Both paper figures look ten times too large, which suggests a fraction was printed as a percentage. The block count and hidden size are the standard ViT-Base architecture, not stated in the paper, so I would ask the authors to confirm.

- For LLaMA2-7B, Table VIII gives rank 1024 on q_proj and v_proj . Over 32 layers that is $32 \times 2 \times 1024 = 65,536$ trainable parameters, about 0.00094% of 7B. I could not find the configuration that yields the "0.00025%" of Section I and Section V, nor the "2600 times" of the Abstract. It would help to say which model, rank, and baseline each figure refers to.

"Without any performance degradation." Section I uses this wording, but Section IVE(d) reports a Table II mean of 82.79% for LoLDU against 88.20% for full fine-tuning, a gap of 5.41 points. The ViT claim that LoLDU beats LoRA by "15.28% and 16.99%" on FGVC and StanfordCars (Section IVC) is interesting, but it also reads as though the LoRA baseline was weak under this recipe. I would soften the Introduction and Conclusion to "competitive with LoRA at a fraction of the parameters," which is what the data seem to support.

Baselines and variance. Section IV states that "Except LoLDU, all results are from prior work" (Table I caption) and that setups were replicated from [9], [14], [57]. Borrowed baselines are common, but they come with different seeds, hardware, and sometimes library versions. For GLUE, LoLDU is the median of five seeds (Section IVA); for ViT, LLaMA2, and Stable Diffusion, a single seed, 42, is listed (Tables VIII, IX, X). Without spread, a difference such as LoLDU beating full fine-tuning on STS-B (Section IVA) or on EuroSAT and Flowers (Section IVC) cannot be separated from noise. I would add standard deviations and at least one re-run of the LoRA baseline in the authors' own pipeline.

Instruction tuning. Section IV.B says LoLDU gives "a slight improvement" over LoRA with "merely 0.05% of the parameters." LoRA uses $r = 64$, but the same paragraph says it updates "all linear layers" and also that it targets W_Q and W_V . On W_Q and W_V alone (32 layers, width 4096, the standard LLaMA2-7B shape), LoRA trains $32 \times 2 \times 64 \times (4096 + 4096) = 33,554,432$ parameters, and LoLDU at rank 1024 (Table VIII) trains 65,536, which is 0.195%, not 0.05%. Please state which modules LoRA used. One seed, three epochs, and differences that the text itself calls slight: I would call this parity, not improvement.

Image generation. Seven concepts (Section IV.D) is a small set. Table VI is not legible as rendered on Qeios, and neither its caption nor Section IV.D mentions any spread across concepts or seeds. Figure 5 is a

single qualitative sequence. The claim of concept acquisition "within ~100 steps" would be more convincing as a curve of CLIP-I or DINO against step, averaged over the seven concepts.

3. Reproducibility

This part is better than average. The code is public (Abstract), the GPU is named (single RTX A6000, 48 GB, Section IV), and Tables VII to X give the learning rate, batch size, epochs or steps, target modules, rank, and alpha for every setting, plus precision and seed for all settings except GLUE (Table VII). Model identifiers are given as `roberta-base`, `vit-b16-224-in21k`, `stable-diffusion-v1-5` with `sd-vae-ft-mse`, and `LLaMA2-7B`.

What I could not find:

- A commit hash or release tag for the repository, so that the numbers can be tied to a specific version of the code.
- Which initialisation of z is used in the main tables. Section III.C lists Constant, Uniform, Normal, and Regular LDU, and Section IV.E(a) says Uniform is best, but Tables VII to X do not state it. This matters: if z_r is re-initialised uniformly on $(-1, 1)$ after the RSM has been computed from the true pivots, the model at step 0 is no longer the pre-trained model. I would state explicitly how Eq. 6 to 8 interact with the heuristic initialisation.
- The pivoting strategy and numerical precision of the decomposition. LU with partial pivoting in `fp16` or `bf16` can be unstable on large matrices; the precision rows of Tables VIII to X refer to training, not to the decomposition.
- Time and memory. Section III.E argues efficiency analytically, but no wall-clock time, peak VRAM, or decomposition time per model is reported. For a paper whose motivation is cost, one table of measured numbers would help a lot.
- The projection operator of Section III.D. Eq. 10 constrains $0 < \sigma \leq 1$ and a Frobenius-norm term whose threshold is not given a value anywhere I could see.
- For INSTRUCTEVAL, the version of the evaluation code and the decoding settings (greedy or sampling, temperature, max new tokens) for MMLU, DROP, BBH, and HumanEval.
- For Table VI, the prompts used, the number of generated images per concept, and the exact DINO and CLIP checkpoints.

4. What specifically to fix

1. **Section I (third paragraph), Section III.A, Section III.C.** Remove the statement that L and U are orthogonal. Only P is. If you want a knowledge-preservation argument, a bound on $\|\Delta W\|$ in terms of σ and z would do the job honestly.
2. **Figure 1 caption vs Section IV.C.** 96.837% and 3.173% do not sum to 100; one is a typo.
3. **Section IV.E(d), Table II.** Recheck the "0.21%" and "6.77%" parameter shares. My count from Table IX gives about 0.0215% and 0.687%, a factor of ten lower in both cases. Please show the count explicitly in a footnote.
4. **Abstract, Section I and Section V.** Name the configuration behind "0.00025%" and "2600 times." From Table VIII I get about 0.00094% for LLaMA2-7B. In Section IV.B, also name the LoRA target modules behind "0.05%."
5. **Tables VII and IX.** Rank 768 is full rank for these models. Say so in Section IV, and add a genuinely low-rank row (for instance $r = 256$) to Tables I and IV.
6. Section I describes RoBERTa-Base as "encoder-decoder." It is encoder-only.
7. **Abstract ("6 natural language understanding datasets") and Section IV.A ("six datasets")** versus Table VII, which lists eight tasks: MNLI, SST-2, MRPC, CoLA, QNLI, QQP, RTE and STS-B. Please reconcile. The same goes for the Abstract's "8 image classification datasets" against the six in Section IV.C and Appendix C, and for Section IV.A applying LoLDU to W_q , W_k , W_v and W_o while Table VII lists query and value only; the 0.0184M in Section IV.A matches the latter ($12 \times 2 \times 768 = 18,432$).
8. **Section IV.E(b), scaling factor.** The paragraph says the gray "without scaling factor" values are in Table IV, but the Table V caption is the one that defines them. It also calls σ "fixed at a value of 1," while Section III.A says σ is dynamically adjusted and Eq. 10 constrains it to (0, 1]. Which one is trained?
9. **Section III.E.** The forward pass of the unmerged DSM costs $O(mr + rn)$ per input vector, not $O(mnr)$; $O(mnr)$ is the cost of forming the product. After merging, inference is $O(mn)$, the same as the base model.
10. **Section II, re-parameterisation paragraph.** The text contrasts $O(mn^2 + n^3)$ for SVD with $O(mn^2 - n^3/3)$ for LDU (repeated in Section III.E). Both share the leading term mn^2 , so the gap is a constant factor. Measured decomposition times would make the advantage concrete.
11. Add seed variance (mean $\pm$ standard deviation over at least three seeds) to Tables III, IV and VI, and re-run LoRA yourselves for at least ViT-Base.

12. **Section I and Section V:** replace "without any performance degradation" and "superiority over traditional fine-tuning methods" with wording that matches the 82.79% vs 88.20% gap reported in Section IVE(d).
13. State the z initialisation used in each main table, and whether the model equals the pre-trained model at step 0.
14. Report wall-clock training time and peak GPU memory for LoLDU and LoRA in one short table.
15. Render Tables I to VI, XI and XII as HTML text on Qeios rather than as images, so that readers and indexers can read the numbers.

5. Recommendation

Minor revisions. The method is simple and cheap, it has been tested on four model families, and the code is public. That is a real contribution. Most of what I ask for is correction and clarification: drop the orthogonality claim, reconcile the parameter percentages, say plainly that the main GLUE and ViT runs are full rank, and soften the "no degradation" wording. The one experimental request, seed variance plus a self-run LoRA baseline, is modest on a single A6000. With those changes I would be glad to see this paper stand.

Declarations

Potential competing interests: No potential competing interests to declare.