MINIMAX UPDATE BALANCING FOR UNIFIED MULTIMODAL PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Unified multimodal pretraining optimizes image understanding (I2T) and generation (T2I) through one shared backbone and one shared optimizer, and balanced task sampling does not balance the optimizer, since under SOAP the realized update serves the two directions unequally. The optimizer averages gradients over accumulation segments rather than balancing tasks directly. As a result, the averaged preconditioned update can be dominated by the segment with the larger norm under the preconditioner's metric. To address this imbalance, we introduce ML-FOP-SOAP, which operates within each gradient-accumulation window. It hierarchically combines consecutive segment averages in a bottom-up manner. At each fold, it computes the difference between the two segment averages, extracts the component orthogonal to the SOAP update, and adds this component back with a coefficient determined by minimax update balancing. This coefficient minimizes the worst-case local surrogate cost across the two segments, preventing either segment from dominating the update. The coefficient has a bounded closed-form solution. It is zero when the two segments are already balanced, equalizes their local surrogate costs when feasible, and otherwise falls back to the standard SOAP update. When the costs can be equalized, the correction follows the preconditioned multiple-gradient descent direction. Importantly, the preconditioner is estimated from the uncorrected mean gradient, keeping the balancing correction separate from preconditioner estimation. On Janus-400M, ML-FOP-SOAP achieves the lowest T2I loss among all tested optimizers and balancing methods while also reducing I2T loss relative to SOAP. It improves the entire I2T–T2I loss frontier across sampling ratios. Its aggregate loss improvement over SOAP is comparable to SOAP's improvement over AdamW, and it matches Muon's performance. Ablation studies show that each design choice matters. Using a fixed coefficient or Euclidean inner products performs worse than SOAP, while applying a single correction over the entire accumulation window captures only part of the improvement. Explicitly contrasting I2T and T2I at every fold degrades performance in both directions, supporting our segment-based approach. Across all three tested architectures, ML-FOP-SOAP achieves the highest mean proportion of updates that benefit both directions, indicating more balanced optimization. Its T2I improvement also increases when scaling from 400M to 1B parameters, where it achieves a 12% improvement in FID-1K.

1 INTRODUCTION

Recent unified multimodal models use a shared generative backbone for both image understanding (I2T) and text-to-image generation (T2I), producing either text or visual tokens (Chameleon Team, 2024; Wang et al., 2026; Wu et al., 2024; Chen et al., 2025; Qu et al., 2024; Xie et al., 2024; Zhou et al., 2024). Although these models may use shared or task-specific visual encoders, tokenizers, and adaptors, both objectives update the same backbone through a shared optimizer (Wu et al., 2024; Chen et al., 2025). The optimizer receives an aggregate gradient from both tasks, but equal sampling weights do not guarantee that the resulting update benefits them equally. I2T and T2I gradients can differ in magnitude, variance, direction, and interaction with the optimizer state, even when their visual pathways are architecturally decoupled (Zheng et al., 2026). This imbalance is

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

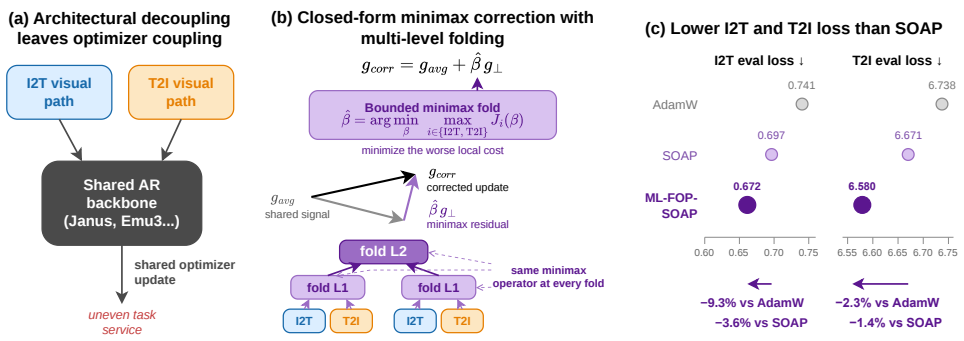

Figure 1: **Overview of ML-FOP-SOAP**, SOAP with a multi-level minimax residual correction (Section 3.4). (a) Architecturally separate image-to-text (I2T) and text-to-image (T2I) pathways are still combined into a single shared optimizer update. (b) ML-FOP-SOAP adds a bounded minimax residual correction at each folding level, chosen to reduce the larger of two local preconditioned surrogate costs, and the operands of each fold are accumulation-segment averages of direction-pure micro-batches (Section 3.4). (c) Reported I2T and T2I validation losses are 3.6% and 1.4%.

related to challenges studied in multi-task optimization (Chen et al., 2017; Sener & Koltun, 2018; Yu et al., 2020; Liu et al., 2021a).

We investigate this problem in the shared backbone of unified multimodal models and distinguish two forms of asymmetry. Statistical asymmetry describes differences in the task gradients themselves, such as the T2I-to-I2T gradient norm or variance ratio. We measure these differences to characterize the two tasks, but do not directly optimize them. Instead, we target update-service asymmetry, which describes how the realized optimizer update benefits each task. For example, an update may align positively with one task's gradient but have non-positive alignment with the other, indicating that it serves only one direction. Under SOAP, such single-direction updates occur more frequently in one direction than the other on both Janus-400M and Emu3-800M (Section 4.3, Appendix E). Our goal is therefore not to equalize gradient statistics, but to produce optimizer updates that serve both objectives more evenly.

The shared optimizer does not receive the two task gradients as a pair. With direction-pure micro-batches and gradient accumulation, what it averages is a sequence of segment gradients whose task composition varies from window to window, and the average is then preconditioned. Loss balancing and multi-task gradient methods reweight objectives or recombine their gradients in Euclidean space (Chen et al., 2017; Sener & Koltun, 2018; Yu et al., 2020; Liu et al., 2021a), and second-order preconditioners supply a better-adapted geometry (Martens & Grosse, 2015; Gupta et al., 2018; Vyas et al., 2024). A preconditioned average is still an average. Once the segments are combined, their disagreement is no longer available to the update rule, and the averaged step favours whichever operand has the larger norm in the preconditioner's metric (Section 3.3). We therefore intervene where the imbalance enters, between consecutive segments inside the window, before they are averaged. At each fold we take the difference between two consecutive segment averages, keep the part orthogonal to the shared step, and add it back with a coefficient chosen so that the two segments' local costs are equal whenever that is possible and bounded otherwise. We build on SOAP (Vyas et al., 2024) and on Fisher-Orthogonal Projection (FOP) (Lu & Armour, 2026), which introduced the orthogonal residual. We call the coefficient rule minimax update balancing and the resulting optimizer ML-FOP-SOAP. Update service denotes the first-order decrease an update gives to an operand, and a fold is one application of the correction to two segment averages. The contributions are as follows.

1. **Quantifying update-service imbalance.** We distinguish statistical asymmetry in task gradients from update-service asymmetry in the realized optimizer update. We show that the averaged SOAP step favours the operand with the larger H_t-norm (Section 3.3). We quantify update service using PDR and Service Bias. Across all three tested architectures,

ML-FOP-SOAP achieves the lowest one-sided Service Bias and the highest mean PDR (Sections 4.3 and 4.3).

2. **Minimax balancing within gradient accumulation.** We introduce a minimax correction that adjusts the coefficient of FOP's orthogonal residual to minimize the larger of two segment-level surrogate costs. Under a shared-curvature approximation with the SOAP step frozen, this optimization admits a bounded, closed-form solution. When the two costs can be equalized, the solution selects the multiple-gradient descent direction under SOAP's inner product while preserving the SOAP component. Otherwise, the coefficient is clipped to limit the correction (Section 3.3). We apply this correction recursively to consecutive segment averages within each accumulation window, allowing segment-level differences to be corrected before they disappear through full-window averaging. Setting all correction coefficients to zero exactly recovers SOAP (Section 3.4).
3. **Performance, and transfer across architectures.** A single full-window correction captures only part of the T2I improvement over SOAP, while recursive folding accounts for the remaining 42–76% across all tested settings. Balancing consecutive segment averages also outperforms explicitly contrasting I2T and T2I at every fold (Table S3). On Janus-400M, ML-FOP-SOAP improves the entire I2T–T2I loss frontier relative to SOAP, achieves the lowest T2I loss among the tested optimizers, and matches Muon's aggregate loss at 5.8% lower throughput than SOAP. Using a fixed correction coefficient or Euclidean inner products instead of SOAP's metric performs worse than SOAP. The improvements in T2I loss and FID-1K also transfer to Emu3-800M and Janus-1B (Sections 4.2–4.3).

2 Related Work

Unified multimodal models. Chameleon and Emu3 model images and text as discrete token sequences (Chameleon Team, 2024; Wang et al., 2026), Janus and Janus-Pro share a language backbone but decouple the visual pathways (Wu et al., 2024; Chen et al., 2025), and TokenFlow, Show-o and Transfusion vary the tokenization and the generative objective (Qu et al., 2024; Xie et al., 2024; Zhou et al., 2024). These recipes balance modalities through model-specific sampling or loss weighting, not through a general rule for balancing I2T and T2I optimization.

Multi-task gradient combination. GradNorm rescales task losses so that gradient magnitudes track training speed (Chen et al., 2017). MGDA takes the minimum-norm element of the convex hull of the task gradients (Sener & Koltun, 2018). CAGrad maximizes the worst-case first-order improvement in a Euclidean trust region around the average gradient (Liu et al., 2021a). IMTL-G equalizes the projections of the combined gradient onto the normalized task gradients (Liu et al., 2021b). FairGrad allocates first-order decreases by an α-fair utility, recovering max–min fairness as $\alpha \to \infty$ (Ban & Ji, 2024). All operate in Euclidean space, and tuned scalarization remains a strong reference point (Xin et al., 2022). Our rule is closest to MGDA, since Proposition 1 shows that for two objectives it reproduces the MGDA direction in SOAP's inner product whenever equalization is feasible.

Curvature-aware optimizers. K-FAC, Shampoo and SOAP supply Kronecker-factored or matrix preconditioners (Martens & Grosse, 2015; Osawa et al., 2023; Gupta et al., 2018; Lin et al., 2026; Vyas et al., 2024). FOP (Lu & Armour, 2026) introduced the orthogonal residual we inherit. It projects under the Fisher metric and sets its coefficient by minimizing a total-loss model of the natural-gradient step, so the coefficient is non-trivial only because the projection and step metrics differ. In SOAP one operator H_t plays both roles and the total-loss criterion degenerates to the SOAP update (Appendix D), so a different criterion is needed.

3 Method

3.1 Problem Formulation and SOAP Geometry

For shared backbone parameters θ with I2T and T2I objectives L_1 and L_2, the correction acts on two operands g_1, g_2, each treated as the gradient of a cost ℓ_i, and all per-operand quantities below are defined with respect to ℓ_i. The operands are the task gradients $\nabla_\theta L_i$ when the two directions

are routed to separate operands (Appendix B), whereas in the reported runs they are the mean gradients of two consecutive segments of the accumulation window (Section 3.4 and Algorithm 1 in Appendix G). The index then names a time segment rather than a modality, and from the second fold on the earlier operand is the corrected running aggregate rather than a raw segment mean. We decompose the operands as

$$g_{\text{avg}} = \tfrac{1}{2}(g_1 + g_2), \qquad g_{\text{diff}} = g_1 - g_2. \tag{1}$$

SOAP maintains structured second-order statistics along both dimensions of each parameter matrix, and in vectorized notation we write its preconditioned update as $P_t(g) = H_t g$, where H_t is the linear map SOAP applies at step t with its eigenbases and second-moment statistics held fixed. In the momentum-free configuration used here (Appendix B), and because those statistics are updated only from the uncorrected mean gradient (Section 3.4), H_t is one and the same symmetric positive-definite operator for every candidate gradient within a step. The inner products below are defined with this map. Standard SOAP applies it to the accumulated mean, $d_{\text{SOAP}} = P_t(g_{\text{avg}})$, so the disagreement carried by g_{diff} never reaches the update. The correction below uses g_{diff} to control how the update is allocated between the two operands, leaving SOAP's preconditioning unchanged.

3.2 Preconditioner-Orthogonal Residual Correction

Adding g_{diff} directly to the shared gradient is undesirable because part of it may be parallel to g_{avg} and would merely rescale the existing direction. Following FOP's construction, we remove that component, taking the projection in the metric of the step so that the residual is orthogonal to the SOAP step $H_t g_{\text{avg}}$ itself,

$$c = \frac{\langle g_{\text{diff}}, H_t g_{\text{avg}} \rangle}{\langle g_{\text{avg}}, H_t g_{\text{avg}} \rangle + \epsilon}, \qquad g_\perp = g_{\text{diff}} - c\, g_{\text{avg}}, \tag{2}$$

and define the corrected gradient and its preconditioned direction as

$$g(\beta) = g_{\text{avg}} + \beta g_\perp, \qquad d(\beta) = P_t\big(g(\beta)\big) = H_t g(\beta). \tag{3}$$

Setting $\beta = 0$ recovers SOAP exactly. Up to the numerical stabilizer ϵ, $g_\perp$ is H_t-orthogonal to g_{avg}.

Equivalently,

$$g(\beta) = \Big[\tfrac{1-\beta c}{2} + \beta\Big] g_1 + \Big[\tfrac{1-\beta c}{2} - \beta\Big] g_2, \tag{4}$$

so the corrected gradient is a data-dependent combination of the two operands with weights summing to $1 - \beta c$. For the coefficient selected below both weights are non-negative and, whenever a correction is applied, sum to more than one (Appendix D.4), so the correction rotates the gradient within the conic hull of g_1 and g_2 without shortening its component along g_{avg}. The coefficient rule below selects the point in this set.

3.3 Closed-Form Minimax Coefficient

Let $\Phi_i(g) = -\langle g_i, g \rangle_{H_t} + \frac{1}{2}\|g\|^2_{H_t}$ be the shared-curvature model of the change in ℓ_i, the loss whose gradient is the operand g_i, produced by the step $H_t g$. Its linear term is the first-order change of ℓ_i, and its quadratic term models the curvature of both losses by H_t^{-1}, so that the SOAP step $H_t g_{\text{avg}}$ is the Newton step of the averaged model. The model is normalized to a unit learning rate. An explicit learning rate η scales the linear terms by η and the quadratic term by η^2, which leaves the equalizing point below unchanged and scales the half-width of the admissible interval by $1/\eta$, with its centre at zero under exact projection (Appendix D). Let $J_i(\beta) = \Phi_i\big(g(\beta)\big)$, so that $J_i(0)$ corresponds to the uncorrected SOAP step. Expanding along the line $g(\beta)$ and using the projection of Eq. (2) gives, exactly and for every ϵ (Appendix D),

$$J_1(\beta) = J_1(0) - \tfrac{\beta}{2} a + \tfrac{\beta^2}{2} b, \qquad J_2(\beta) = J_2(0) + \tfrac{\beta}{2} a + \tfrac{\beta^2}{2} b, \tag{5}$$

with

$$a = \langle g_{\text{diff}}, H_t g_\perp \rangle, \qquad b = \langle g_\perp, H_t g_\perp \rangle, \qquad \Delta = J_2(0) - J_1(0) = \langle g_{\text{diff}}, H_t g_{\text{avg}} \rangle. \tag{6}$$

Here Δ is the signed local imbalance between the two operands, a is the rate at which the correction changes that imbalance, and b is the shared quadratic cost both directions pay for moving along $g_\perp$. Because H_t is symmetric, the imbalance reads

$$\Delta = \tfrac{1}{2}\big(\|g_1\|^2_{H_t} - \|g_2\|^2_{H_t}\big), \qquad \langle g_i, H_t g_{\text{avg}} \rangle = \tfrac{1}{2}\big(\|g_i\|^2_{H_t} + \langle g_1, H_t g_2 \rangle\big),$$

so the uncorrected SOAP step serves each operand in proportion to its squared H_t-norm, the cross term being common to both. Local balance, $\Delta = 0$, is equality of the two H_t-norms, and the service gap after a correction β is $J_2(\beta) - J_1(\beta) = \Delta + a\beta$. We select

$$\hat{\beta} = \arg\min_{\beta} \max_{i} J_i(\beta), \tag{7}$$

which reduces to the one-dimensional convex problem $\min_\beta \frac{b}{2}\beta^2 + \frac{1}{2}|\Delta + a\beta|$ and has the bounded closed form

$$\hat{\beta} = \mathrm{clip}\left(-\frac{\Delta}{a},\, -\frac{a}{2b},\, \frac{a}{2b}\right). \tag{8}$$

The rule has three regimes. When $\Delta \approx 0$ the two operands are locally balanced, $\hat{\beta} \approx 0$, and the update is the SOAP update. When the equalizing point $-\Delta/a$ lies inside the interval, the correction scales with the measured imbalance and equalizes the two surrogate costs. When it does not, the quadratic cost dominates first and the solution clips to the boundary that gives the largest admissible improvement to the worse side. Equation (8) is the exact minimizer of $\max_i J_i(\beta)$ for every value of the stabilizer ϵ. A finite ϵ enters only through the interval, because the projection leaves the residual $q = \langle g_{\mathrm{avg}}, H_t g_\perp \rangle = \epsilon\Delta/(D+\epsilon)$ with $D = \|g_{\mathrm{avg}}\|^2_{H_t}$, and $a = b + cq \geq b$, so the half-width $a/(2b)$ exceeds $\frac{1}{2}$ by a term of order ϵ/D. Under exact projection $a = b$ and the interval is exactly $[-\frac{1}{2}, \frac{1}{2}]$ (Appendix D). The surrogate coincides with a classical multi-objective criterion. Since $J_i(\beta) = \Phi_i\big(g(\beta)\big)$ by definition, Eq. (8) minimizes $\max_i \Phi_i$ over the line $\{g_{\mathrm{avg}} + \beta g_\perp\}$ (Appendix D.4). Minimizing the same objective over *all* g is the min–max characterization of multiple-gradient descent (Sener & Koltun, 2018), whose solution by convex duality is the minimum-norm element of the convex hull of the task gradients, here in the H_t inner product,

$$g^\star = \arg\min_{g \in \mathrm{conv}\{g_1, g_2\}} \|g\|_{H_t}. \tag{9}$$

The minimax coefficient and $g^\star$ are related as follows.

Proposition 1. *Assume exact projection ($\epsilon = 0$, so $a = b$), $D = \|g_{\mathrm{avg}}\|^2_{H_t} > 0$ and $b > 0$. Let $\hat{g} = g(\hat{\beta})$ and $\kappa = bD/(bD + \Delta^2) \in (0, 1]$.*

- *(i) If $\Delta = 0$, then $\hat{g} = g^\star = g_{\mathrm{avg}}$.*
- *(ii) If $|\Delta| < b/2$ (the active regime), then $\hat{g} = g^\star/\kappa$, so the two rules select the same direction.*
- *(iii) In every regime, $\langle \hat{g},\, g_{\mathrm{avg}} \rangle_{H_t} = D \geq \langle g^\star, g_{\mathrm{avg}} \rangle_{H_t}$, with equality only if $g^\star = g_{\mathrm{avg}}$.*
- *(iv) If $|\Delta| \geq b/2$ (the saturated regime), then $\hat{g} = g_{\mathrm{avg}} - \frac{1}{2}\mathrm{sign}(\Delta)\, g_\perp$, whose angle θ to g_{avg} in the H_t geometry satisfies $\tan\theta = \frac{1}{2}\sqrt{b/D}$, whereas $g^\star$ can rotate further, up to a single task gradient.*

Under the assumptions of the proposition, the minimax coefficient thus selects the direction of preconditioned multiple-gradient descent through FOP's residual parameterization (ii), keeps the component along the SOAP direction fixed instead of contracting it by the factor κ of the minimum-norm point (iii), and bounds the rotation when equalization is infeasible (iv). The SOAP-MGDA baseline of Section 4.1 differs from Eq. (8) in three respects. It forms its simplex weights from Euclidean inner products before preconditioning, it returns the minimum-norm point, which may rotate up to a single task gradient, and it is not bounded as in (iv).

3.4 Multi-Level Folding and Optimizer-State Isolation

A single correction after the window has been averaged acts only on the full-window mean, so we apply the same operator recursively inside the window (Algorithm 1, Appendix G). With K micro-steps and dyadic checkpoints $L_j \in \{1, 2, 4, \ldots, K\}$, fold j takes $g_1 = z_{j-1}$, the corrected aggregate over micro-steps $1{:}L_{j-1}$, and $g_2 = \bar{g}_{L_{j-1}+1:L_j}$, the average of the next, equally long segment, and the final direction is $P_t(z_J)$. Here J_2 is the surrogate for the later segment's mean loss and J_1 that of the corrected aggregate, which differs from the earlier segment's only through the bounded residuals already folded in, so $\hat{\beta}$ equalizes these segment-level costs rather than the I2T and T2I losses themselves. With direction-pure micro-batches the segment difference carries

a modality contrast in proportion to the difference in task composition, and forcing a pure I2T-versus-T2I contrast at every fold instead degrades both directions (Table S3). When every fold coefficient is zero the recursion returns the accumulated mean, hence SOAP. The SOAP statistics that define H_t, including the eigenbasis and second-moment statistics, are updated from the uncorrected accumulated mean, so a correction never feeds back into preconditioner estimation. Within one fold the correction only redistributes first-order descent, since by Eq. (5) $[J_1(\beta) - J_1(0)] + [J_2(\beta) - J_2(0)] = \beta^2 b$, and the end-to-end effect of the recursion is what Tables 1 and S3 measure.

4 Experiments

4.1 Setup

We use Janus-400M (Chen et al., 2025) trained on LLaVA-CC12M+CC3M image–caption pairs (Changpinyo et al., 2021; Sharma et al., 2018) as the main setting, where each pair supplies one I2T and one T2I instance, sampled in equal proportion, with a 10B-token budget and seeds 0, 1 and 2. AdamW, Muon, SOAP and ML-FOP-SOAP are also trained at 25:75 and 75:25 sampling ratios with the same three seeds (Appendix F), and Emu3-800M (Wang et al., 2026) and Janus-Pro-1B-official (Chen et al., 2025) (Janus-1B in the tables) at 16B and 20B tokens, one run per configuration, at roughly $20\times$ tokens per parameter (Hoffmann et al., 2022). Baselines are AdamW (Loshchilov & Hutter, 2017), Muon (Jordan et al., 2024), GradNorm, CAGrad and MGDA on AdamW (Chen et al., 2017; Liu et al., 2021a; Sener & Koltun, 2018), SOAP (Vyas et al., 2024), and CAGrad and MGDA on SOAP, while single-shot FOP-SOAP (Lu & Armour, 2026) separates the residual correction from its multi-level placement. Generation is scored by OpenCLIP cosine similarity (Cherti et al., 2023) and FID (Heusel et al., 2017), understanding by MMKC-VS (Jia et al., 2026), CIDEr (Vedantam et al., 2015) and BLEU-4 (Papineni et al., 2002), the latter two on the first sentence of each generated caption (CIDEr-first, BLEU-first). Diagnostics over the 6–8B-token window comprise two descriptive ratios of T2I to I2T gradient statistics and two update-service measures, PDR and Service Bias (Section 4.3, Appendix C), and training configurations are in Appendix B.

4.2 Understanding and Generation Quality

Table 1: Validation losses and optimizer diagnostics for Janus-400M (mean $\pm$ standard deviation across reported training trajectories). Bold marks the best mean, and in the two ratio columns the entry closest to parity. Arrows give the direction of preference, and the ratio columns are descriptive gradient statistics with parity at one (Section 4.3). Single-shot FOP-SOAP applies one correction to the full-window mean, and its costs are in Table S5.

Method	Eval loss ↓	Eval I2T ↓	Eval T2I ↓	Norm ratio (parity 1)	Var. ratio (parity 1)	PDR ↑	Service bias ↓
Standard optimizers							
AdamW	3.740±0.008	0.741±0.004	6.738±0.017	0.782±0.037	0.220±0.019	0.228±0.033	0.079±0.054
Muon	**3.622**±0.011	**0.568**±0.001	6.675±0.023	1.235±0.222	0.193±0.096	0.246±0.051	0.070±0.071
Task balancing on AdamW							
AdamW-GradNorm	3.851±0.017	0.657±0.046	7.045±0.015	0.865±0.439	0.372±0.174	0.222±0.026	0.050±0.034
AdamW-CAGrad	3.819±0.003	0.696±0.004	6.942±0.001	0.749±0.073	0.307±0.189	0.206±0.008	0.089±0.057
AdamW-MGDA	4.292±0.014	0.714±0.019	7.869±0.008	0.777±0.090	0.363±0.115	0.232±0.035	0.034±0.016
SOAP-based controls							
SOAP	3.684±0.009	0.697±0.007	6.671±0.010	0.689±0.029	0.304±0.049	0.262±0.062	0.071±0.053
SOAP-CAGrad	3.821±0.030	0.614±0.067	7.027±0.043	0.490±0.057	0.446±0.184	0.223±0.077	0.040±0.043
SOAP-MGDA	4.149±0.013	0.776±0.010	7.523±0.013	0.568±0.047	0.303±0.222	0.217±0.016	0.050±0.021
FOP-SOAP single-shot	3.649±0.010	0.680±0.007	6.618±0.014	0.644±0.024	0.509±0.142	0.264±0.047	0.026±0.026
ML-FOP-SOAP	3.625±0.012	0.672±0.010	**6.580**±0.018	**1.064**±0.013	**0.720**±0.185	**0.283**±0.013	**0.009**±0.005

Table 1 reports the validation losses in its first three columns. ML-FOP-SOAP has the lowest T2I loss (6.580), 0.091 below SOAP and 0.095 below Muon, and its aggregate loss (3.625) ties Muon's (3.622) within seed variation and lies 0.059 below SOAP's. Muon has the lowest I2T loss (0.568), and ML-FOP-SOAP's (0.672) is 0.025 below SOAP's. For scale, SOAP improves on AdamW by 0.056 in aggregate and 0.067 in T2I loss, and ML-FOP-SOAP improves on SOAP by 0.059 and 0.091, at a throughput cost of 5.8% against SOAP's 1.6% over AdamW (Section 4.2). A single correction of the full-window mean (single-shot row) already lowers both directional losses, and the

Table 2: Downstream generation and understanding results for Janus-400M. Entries are cross-seed mean ± standard deviation, and bold marks the best mean. FID-1K and CLIP cosine use a fixed set of 1,000 prompts and a fixed generation seed, so the standard deviations reflect training seeds.

Method	Generation		Understanding		
	CLIP ↑	FID-1K ↓	MMKC-VS ↑	CIDEr-first ↑	BLEU-first ↑
AdamW	0.172±0.006	191.117±8.182	0.265±0.012	0.377±0.006	0.115±0.002
Muon	0.165±0.009	190.520±5.023	0.234±0.014	**0.384**±0.002	0.117±0.003
AdamW-GradNorm	0.166±0.005	217.092±8.483	0.283±0.039	0.381±0.006	0.116±0.003
AdamW-CAGrad	0.164±0.001	193.674±8.351	0.238±0.048	0.376±0.005	0.113±0.001
AdamW-MGDA	0.163±0.014	193.068±2.551	0.255±0.024	0.380±0.011	0.115±0.001
SOAP	0.174±0.006	179.369±6.923	0.240±0.031	0.380±0.003	0.118±0.002
SOAP-CAGrad	0.169±0.003	198.398±9.859	0.268±0.010	0.383±0.008	**0.120**±0.003
SOAP-MGDA	0.170±0.005	191.266±5.340	0.259±0.019	0.375±0.005	0.116±0.001
ML-FOP-SOAP	**0.177**±0.007	**174.838**±2.287	**0.286**±0.024	0.381±0.002	**0.120**±0.003

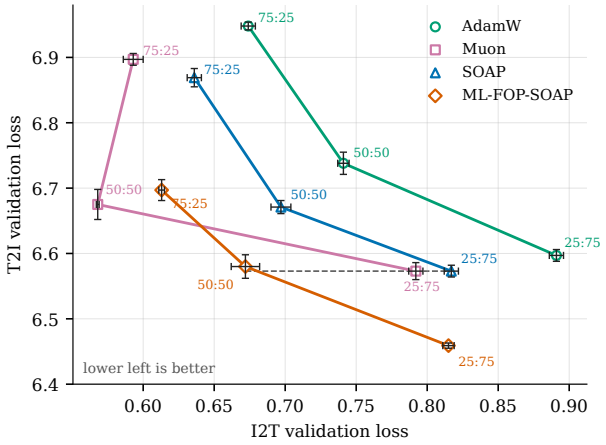

Figure 2: I2T–T2I loss frontier on Janus-400M across I2T:T2I sampling ratios (25:75, 50:50 and 75:25, lower left is better). Points are means and error bars standard deviations over three seeds at every ratio, and bars shorter than a marker lie inside it. The dashed guide marks the T2I loss of SOAP at 25:75, which ML-FOP-SOAP at 50:50 approaches within 0.007 with I2T loss 0.145 lower.

multi-level placement lowers them further, T2I by 0.038 and I2T by 0.008, with the coefficient rule, preconditioner and protocol held fixed. Two controls in Table S2 isolate the rule and the geometry. With a fixed weak-side coefficient or with Euclidean inner products the same residual falls below SOAP in both directions, whereas the single-shot correction already lowers Service Bias toward ML-FOP-SOAP's value.

Figure 2 places the four base configurations in the plane of I2T against T2I loss at 25:75, 50:50 and 75:25 sampling ratios (Appendix F). At every ratio ML-FOP-SOAP lowers T2I loss relative to SOAP by 0.091 or more without raising I2T loss, so its frontier lies inside SOAP's along its whole length, and no change of mixture moves SOAP onto it. At 50:50, ML-FOP-SOAP's T2I loss lies within 0.007 of the value SOAP attains only with 75% T2I data (6.580 against 6.573), with I2T loss 0.145 lower (0.672 against 0.817). Its aggregate loss is the lowest of the four at 25:75 and 75:25, ties Muon at 50:50, and varies by 0.030 across ratios against 0.069 for SOAP, 0.071 for AdamW and 0.123 for Muon. The ML-FOP-SOAP and Muon frontiers cross, ML-FOP-SOAP inside on the generation side and Muon on the I2T side.

Table 2 reports downstream quality. ML-FOP-SOAP has the best mean on four of the five metrics. On FID-1K it leads Muon and every task-balancing baseline by 15 or more points, lies within seed

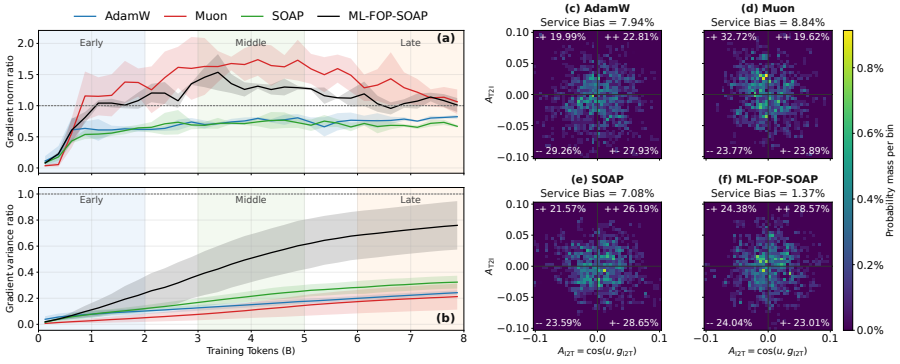

Figure 3: Gradient-statistical asymmetry and update-service geometry. (a,b) T2I-to-I2T gradient norm and variance ratios across training, with dashed lines marking statistical parity. (c–f) Probability-normalized realized-update alignment distributions over 6–8B tokens, with quadrant proportions and Service Bias annotations. Quadrant proportions pool all logged 6–8B update events.

variation of SOAP (174.8 against 179.4), and has the smallest standard deviation across seeds (2.3) in the column. On MMKC-VS it has the best mean (0.286) and ties AdamW-GradNorm, and on BLEU-first it ties SOAP-CAGrad. CLIP cosine and CIDEr-first do not separate the methods at three seeds, all means lying within 0.014 and 0.009 of each other.

Muon and ML-FOP-SOAP reach the same aggregate loss by opposite allocations, since Muon's I2T loss is 15% below ML-FOP-SOAP's while its T2I loss is no better than SOAP's, and ML-FOP-SOAP's T2I loss is 1.4% below Muon's with an I2T loss below SOAP's. The understanding benchmarks do not resolve Muon's I2T-loss advantage at three seeds (CIDEr-first lead 0.003, MMKC-VS 0.052 lower), whereas FID-1K resolves the T2I side (15.7 behind ML-FOP-SOAP, about five pooled standard errors). Changing the mixture does not reproduce the pattern, since with 75% T2I data Muon's T2I loss (6.573) only matches ML-FOP-SOAP's at 50:50, its FID-1K rises to 245, and its aggregate loss varies four times more across mixtures (Appendix F). Muon is thus the strongest choice when I2T loss alone is the objective, and the clearest instance of one-sided service in our tables (Section 4.3).

On 252 training windows the closed-form coefficient of Eq. (8) matches an exhaustive grid search over the same surrogate to within 1.06×10^{-5}. Its worse-side surrogate benefit over SOAP stays non-negative across imbalance regimes, whereas a fixed weak-side-vertex rule overshoots under light-to-moderate imbalance (Appendix D.5, Figure S1), which is also the ordering the two rules show end to end in Table S2.

Against both training tokens and estimated model time, ML-FOP-SOAP stays below SOAP in aggregate validation loss (Figure S2 in Appendix E), and after equal model time it has processed 9.42B tokens against 10.00B for SOAP and reaches a lower loss, while Muon follows the lowest aggregate-loss trajectory. ML-FOP-SOAP runs at 94,391 tokens/s, 5.8% below SOAP and 1.7% below single-shot FOP-SOAP, and uses 1.952 GB more peak memory than SOAP, of which 0.400 GB comes from folding beyond the single-shot correction. Appendix E reports the other architectures.

4.3 Gradient Statistics and Update-Service Diagnostics

Equal sampling fixes how often each direction is seen, not the gradients the optimizer receives. In Figure 3(a,b) the T2I-to-I2T norm ratio sits below parity for AdamW and SOAP and above it for Muon and ML-FOP-SOAP, and the cumulative variance ratio stays below one for every method. These ratios are descriptive, since they scale with the units of each loss, and parity is not a target. Panels (c–f) show the realized update's alignment with the two task gradients, where PDR is the fraction of updates positively aligned with both, and Service Bias is the absolute difference between the rates at which only I2T or only T2I is served.

Section 3.3 predicts the direction of the gain. The averaged SOAP step favours the operand with the larger H_t-norm, and under SOAP the Euclidean norm ratio of the task gradients in the probe space puts T2I on the smaller side (0.689), so equalizing service should move T2I loss first. The losses

Table 3: Cross-architecture transfer at fixed token budgets of 16B tokens for Emu3-800M and 20B tokens for Janus-Pro-1B, one run per row. Comparisons are within architecture.

Architecture	Method	Directional evaluation		Generation		Understanding	
		I2T loss ↓	T2I loss ↓	CLIP ↑	FID-1K ↓	CIDEr-first ↑	BLEU-first ↑
Emu3-800M	AdamW	1.022	6.611	0.148	207.2	0.089	0.148
Emu3-800M	Muon	1.009	6.434	0.152	197.5	0.141	0.175
Emu3-800M	SOAP	1.011	6.459	0.147	201.6	0.131	0.168
Emu3-800M	FOP-SOAP single-shot	1.008	6.449	0.152	195.6	0.146	0.174
Emu3-800M	ML-FOP-SOAP	**1.001**	**6.418**	**0.154**	**190.3**	**0.155**	**0.179**
Janus-1B	AdamW	0.729	6.316	0.173	185.8	0.406	0.147
Janus-1B	Muon	**0.609**	6.343	0.168	194.6	**0.419**	0.155
Janus-1B	SOAP	0.678	6.232	0.176	175.9	0.403	0.154
Janus-1B	FOP-SOAP single-shot	0.672	6.189	0.174	169.7	0.409	0.158
Janus-1B	ML-FOP-SOAP	0.663	**6.115**	**0.187**	**155.0**	0.412	**0.161**

confirm this, as T2I is the column in which ML-FOP-SOAP leads every baseline most clearly. The norm ratio of task-pure probe gradients, which the rule does not optimize, moves from 0.689 under SOAP to 1.064 under ML-FOP-SOAP. The two kinds of asymmetry can also point in opposite directions, as under Muon the norm ratio exceeds parity (1.235) while its losses favour I2T (Section 4.2). ML-FOP-SOAP has the highest mean PDR (0.283) and the lowest Service Bias (0.009). At three seeds the bias reduction is about two pooled standard errors while the PDR gain lies within SOAP's seed spread, and the same ordering holds on Emu3-800M and Janus-1B (Table S7).

Raw I2T–T2I gradient cosines have means within 0.012 of zero for every method, and the fraction of negative cosines (0.36 to 0.71 across methods, Appendix C) does not order the methods, whereas the four best methods by T2I loss and FID-1K (Tables 2, 1 and S2) are also the four best by PDR, in the same order. The diagnostics thus point to asymmetric update service, not conflict, as what separates the methods.

Table 3 reports one run per configuration, and Table S6 in Appendix E calibrates its gaps against the run-to-run spread of the Janus-400M seeds. Relative to SOAP, ML-FOP-SOAP is on the better side of all six metrics in all three settings, and folding accounts for 42–76% of the T2I-loss gain over SOAP in every setting. The Janus-1B gaps in T2I loss, FID-1K and CIDEr-first are 5.7, 2.9 and 2.5 times the 400M spread, and the generation gaps are larger than at 400M, as FID-1K improves by 11.9% over SOAP and CLIP cosine by 0.011. Against Muon, ML-FOP-SOAP leads on T2I loss and FID-1K in all three settings, Muon's advantage being confined to I2T loss and CIDEr-first on the Janus family, and on Emu3-800M ML-FOP-SOAP leads on all six metrics.

5 Conclusion

Balanced sampling ensures that I2T and T2I are observed equally often, but does not ensure that they benefit equally from the shared preconditioned update. Moreover, the frequency of gradient conflicts alone does not explain the performance differences between optimizers. Instead, we focus on update-service asymmetry, the degree to which the realized update benefits the two directions unevenly. To address this imbalance, ML-FOP-SOAP applies minimax update balancing at every fold within the gradient-accumulation window. At each fold, it adjusts the coefficient of FOP's orthogonal residual using a bounded closed-form solution. The coefficient equalizes the local surrogate costs of two consecutive segments whenever feasible and reduces to the standard SOAP update when the segments are already balanced. This approach balances the segment averages that the optimizer actually combines, rather than explicitly balancing task gradients. Across all three tested architectures, it produces updates that serve I2T and T2I more evenly. On Janus-400M, ML-FOP-SOAP improves the entire I2T–T2I loss frontier relative to SOAP and matches Muon's aggregate loss at 5.8% lower throughput than SOAP. Its improvements in T2I loss and FID-1K also hold on Emu3-800M and become larger on Janus-1B. Beyond two objectives, the same formulation generalizes to a small convex minimax problem over the H_t-orthogonal residual subspace.

REFERENCES

Hao Ban and Kaiyi Ji. Fair resource allocation in multi-task learning. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 2715–2731. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/ban24a.html.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models, 2024. URL https://arxiv.org/abs/2405.09818.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3558–3568, 2021. doi: 10.1109/CVPR46437.2021.00356.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling, 2025. URL https://arxiv.org/abs/2501.17811.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks, 2017. URL https://arxiv.org/abs/1711.02257.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization, 2018. URL https://arxiv.org/abs/1802.09568.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030, 2022.

Yifan Jia, Yuntao Du, Kailin Jiang, Yuyang Liang, Qihan Ren, Yi Xin, Rui Yang, Fenze Feng, Mingcai Chen, Hengyang Lu, Haozhe Wang, Qian Li, Xiaoye Qu, and Dongrui Liu. Benchmarking multimodal knowledge conflict for large multimodal models. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence*, volume 40, pp. 22283–22291, 2026. doi: 10.1609/aaai.v40i27.39385. URL https://doi.org/10.1609/aaai.v40i27.39385.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Wu Lin, Scott C. Lowe, Felix Dangel, Runa Eschenhagen, Zikun Xu, and Roger B. Grosse. Understanding and improving shampoo and soap via kullback-leibler minimization, 2026. URL https://arxiv.org/abs/2509.03378.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning, 2021a. URL https://arxiv.org/abs/2110.14048.

Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b. URL https://openreview.net/forum?id=IMPnRXEWpvr.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017. URL https://arxiv.org/abs/1711.05101.

Yishun Lu and Wesley Armour. Fisher-orthogonal projection for natural gradient descent in large batch training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24115–24123, 2026.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature, 2015. URL https://arxiv.org/abs/1503.05671.

Kazuki Osawa, Satoki Ishikawa, Rio Yokota, Shigang Li, and Torsten Hoefler. Asdl: A unified interface for gradient preconditioning in pytorch, 2023. URL https://arxiv.org/abs/2305.04684.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K. Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation, 2024. URL https://arxiv.org/abs/2412.03069.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization, 2018. URL https://arxiv.org/abs/1810.04650.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 2556–2565, 2018. doi: 10.18653/v1/P18-1238.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4566–4575, 2015.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. Soap: Improving and stabilizing shampoo using adam, 2024. URL https://arxiv.org/abs/2409.11321.

Xinlong Wang, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Zhen Li, Yuqi Wang, et al. Multimodal learning with next-token prediction for large multimodal models. *Nature*, pp. 1–7, 2026.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation, 2024. URL https://arxiv.org/abs/2410.13848.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation, 2024. URL https://arxiv.org/abs/2408.12528.

Derrick Xin, Behrooz Ghorbani, Justin Gilmer, Ankush Garg, and Orhan Firat. Do current multi-task optimization methods in deep learning even help? In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/580c4ec4738ff61d5862a122cdf139b6-Abstract-Conference.html.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning, 2020. URL https://arxiv.org/abs/2001.06782.

Dian Zheng, Manyuan Zhang, Hongyu Li, Kai Zou, Hongbo Liu, Ziyu Guo, Kaituo Feng, Yexin Liu, Ying Luo, and Hongsheng Li. Aia: Rethinking architecture decoupling strategy in unified multimodal model, 2026. URL https://arxiv.org/abs/2511.22663.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model, 2024. URL https://arxiv.org/abs/2408.11039.

Supplementary Material

A Use of Large Language Models

In this work, we used generative AI tools (Claude, GPT, ChatGPT and Gemini) for three tasks that require disclosure. Claude Fable 5 checked the mathematical derivations in Section 3 and Appendix D, including counterexamples and boundary cases, and suggested corrections to them. The tools computed derived statistics (differences, standard errors) from our logged results and assisted in their interpretation, and they gave feedback on which controls and ablations to report. We have not used generative AI tools to generate data or to design the experiments, and translation and dataset cleaning are not applicable to this work. Additionally, we used generative AI tools to write and debug experiment, plotting and LaTeX code, to edit figures, and to draft, restructure and polish the text, which supports the writing of one author who has a learning difficulty. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

B Training and Evaluation Details

B.1 Data, Task Mixture, and Model Configuration

We train only on LLaVA-CC12M+CC3M image–caption pairs, each formatted in both directions (I2T and T2I), and no pure-text, class-name or instruction data is used. After filtering, the training set has 12,293,925 pairs, and a fixed, seeded 1% of CC12M (95,317 pairs) is held out for validation. The I2T:T2I ratio is imposed per micro-step, so every micro-step is direction-pure, and accumulated micro-batch losses are weighted equally, without a direction-specific loss weight. Table S1 lists the architectures. The Janus variants keep the SigLIP understanding encoder, the VQ tokenizer and the VQ decoder of Janus-Pro-1B frozen and train the shared backbone, both adaptors and the image-prediction head, while adaptors, generation head and generation embeddings start from scratch before the common warm-start phase. T2I predicts 16,384-way image tokens over 576 positions and I2T predicts caption tokens, so the two losses live on different scales.

Table S1: Model configurations. Janus-400M and Emu3-800M are custom scaled variants, and the official architectures are listed for reference.

Model	Status	Params.	Layers	Hidden / FFN	Q heads / KV / context
Janus-400M	custom	$\approx$ 400M	14	1024 / 2816	8 / 8 / 4096
Janus-Pro-1B-official	official	$\approx$ 1B	24	2048 / 5504	16 / 16 / 4096
Emu3-800M	custom	$\approx$ 800M	32	1024 / 4096	16 / 16 / 5120
Emu3-8B-official	official	$\approx$ 8B	32	4096 / 14336	32 / 8 / 131072

B.2 Optimization, Checkpoint Selection, and Compute

All optimizers start from the same seed-matched warm-start checkpoint of each architecture and share the 50:50 mixture, data order, global batch of 512 sequences (gradient accumulation 4, with per-device batch 64, 16 and 32 for Janus-400M, Janus-Pro-1B-official and Emu3-800M), cosine schedule with 5% warm-up, decoupled weight decay 0.1, gradient-norm clipping 1.0, and seeds 0, 1 and 2. Following SOAP (Vyas et al., 2024), each method is swept over the half-decade grid $\{0.1, 0.0316, 0.01, 0.00316, 0.001, 0.000316\}$, and we keep the learning rate with the lowest mean held-out aggregate loss (Appendix B.4) at the fixed-token endpoint and read every reported number from that endpoint. Training used nodes of eight AMD MI300X GPUs (two for Janus-400M, eight for Janus-Pro-1B-official, four for Emu3-800M) with pure data parallelism, bfloat16 compute, and float32 optimizer state and FOP inner products.

B.3 ML-FOP-SOAP TRAINING CONFIGURATION

Task routing is sampled once per micro-step and broadcast, so micro-batches are direction-pure and the optimizer sees only the gradient sequence. The reported runs use no paired-task routing, so fold operands are segment averages of random task composition (Section 3.4). With $K = 4$ we keep cumulative-gradient snapshots at the dyadic boundaries $\{1, 2, 4\}$ and fold bottom-up, two folds per window, and each FOP inner product is one float32 scalar over a parameter group. The coefficient is the closed form of Eq. (8) with stabilizer $\epsilon = 10^{-12}$, degeneracy threshold $\tau_{\text{deg}} = 10^{-12}$, absolute cap $\beta_{\text{abs}} = 0.50$ (Table S8) and a 1,000-step dormant period of plain SOAP. State isolation is on throughout, so SOAP curvature, eigenbases and second-moment statistics are updated from the uncorrected accumulated mean while the search direction uses the corrected gradient. SOAP preconditions two-dimensional tensors of block dimension at least 1024, with preconditioner updates every 100 steps, damping 0.05, no momentum and decoupled weight decay, and one-dimensional parameters use a separate Adam group at learning rate 3×10^{-4}.

Table S2 separates the three design decisions of the correction, the coefficient rule, the placement and the geometry, with the residual, the preconditioner with state isolation, the warm start and the protocol held fixed. Table S3 compares task-pure fold operands with the random-composition operands of the reported runs.

Table S2: Coefficient rule, placement and geometry on Janus-400M at 10B tokens. SOAP, single-shot minimax and ML-FOP-SOAP rows are three-seed means (Tables 2 and 1), and the WSV and Euclidean rows are single runs (seed 0). WSV uses the fixed weak-side-vertex coefficient of Appendix D, and Euclidean computes projection and coefficient with $H_t = I$ before the unchanged SOAP update.

Method	Coefficient	Placement	Geometry	I2T loss ↓	T2I loss ↓	MMKC-VS ↑	CIDEr ↑	Gen. CLIP ↑	FID-1K ↓	PDR ↑	Service bias ↓
SOAP	–	–	–	0.697	6.671	0.240	0.380	0.174	179.369	0.262	0.071
FOP-SOAP	WSV (fixed)	single-shot	H_t	0.803	6.912	0.225	0.379	0.173	194.036	0.225	0.055
FOP-SOAP	minimax	single-shot	H_t	0.680	6.618	0.270	0.380	0.176	176.512	0.264	0.026
ML-FOP-SOAP	minimax	multi-level	H_t	0.672	6.580	0.286	0.381	0.177	174.838	0.283	0.009
ML-FOP-SOAP	minimax	multi-level	Euclidean	0.712	6.898	0.232	0.379	0.172	191.243	0.235	0.048

Table S3: Routing control on Janus-400M, comparing task-pure I2T and T2I fold operands (paired) against the random-composition segment averages of the reported runs (unpaired), with the same micro-batches, fold depth and compute.

Method (routing)	I2T loss ↓	T2I loss ↓	PDR ↑	Service bias ↓
SOAP (paired)	0.704	6.689	0.257	0.076
SOAP (unpaired)	0.697	6.671	0.262	0.071
ML-FOP-SOAP (paired)	0.893	6.760	0.253	0.060
ML-FOP-SOAP (unpaired)	0.672	6.580	0.283	0.009

B.4 EVALUATION PROTOCOL

Diagnostics are computed in the random-projection space of one fixed shared-backbone layer, with the same projection for every optimizer, and probes are refreshed every 50 steps and aggregated over the 6–8B-token window (Appendix C). Validation losses are token means over the valid target tokens of each direction on the held-out bidirectional pairs, and the reported aggregate loss is the unweighted mean of the two directional losses. Downstream metrics use the same fixed-token checkpoint, prompts, decoding and evaluation seed for every method. Generation uses 1,000 fixed prompt–reference pairs, one image per prompt by ancestral sampling (temperature 1.0), classifier-free guidance 5.0 and generation seed 0, with CLIP cosine from OpenCLIP ViT-L/14 (OpenAI weights) and FID-1K from clean-fid on the same pairs. MMKC-VS is the full 147-item image-semantics split of MMKC-Bench (Jia et al., 2026), four-way multiple choice scored by length-normalized likelihood with PMI calibration. Captions are greedy with the training template, and CIDEr and BLEU-4 use an in-house tokenizer, so absolute values differ from published numbers. Tables report mean and sample standard deviation over three seeds.

C Diagnostic Metric Definitions

On shared-backbone parameters, let g_{I2T} and g_{T2I} be the task gradients on matched batches and $u_t = -\Delta\theta_t$ the realized update at logging point t. The ratio columns are

$$R_{\mathrm{norm}} = \frac{\|g_{\mathrm{T2I}}\|}{\|g_{\mathrm{I2T}}\| + \epsilon}, \qquad R_{\mathrm{cov}} = \frac{\mathrm{Tr}\,\mathrm{Cov}(g_{\mathrm{T2I}})}{\mathrm{Tr}\,\mathrm{Cov}(g_{\mathrm{I2T}}) + \epsilon}, \tag{10}$$

both compared with parity at one. With $A_{\mathrm{I2T}}(t) = \cos(u_t, g_{\mathrm{I2T},t})$ and $A_{\mathrm{T2I}}(t) = \cos(u_t, g_{\mathrm{T2I},t})$,

$$\mathrm{PDR} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}[A_{\mathrm{I2T}}(t) > 0 \ \wedge \ A_{\mathrm{T2I}}(t) > 0], \qquad \mathrm{PDR} \uparrow, \tag{11}$$

and, with the one-sided rates

$$p_{+-} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}[A_{\mathrm{I2T}}(t) > 0,\ A_{\mathrm{T2I}}(t) \le 0], \qquad p_{-+} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}[A_{\mathrm{I2T}}(t) \le 0,\ A_{\mathrm{T2I}}(t) > 0], \tag{12}$$

we report

$$S_{\mathrm{bias}} = |p_{-+} - p_{+-}|, \qquad S_{\mathrm{bias}} \downarrow, \tag{13}$$

which is zero when one-sided service is symmetric. Covariances are estimated from the probes inside the window, and alignments use the realized update, so every optimizer is judged in the geometry it actually moves in. Service bias is read together with PDR, since both alignments can be negative. Table S4 reports the raw gradient cosine $C_{\mathrm{grad}} = \cos(g_{\mathrm{I2T}}, g_{\mathrm{T2I}})$ and the conflict rate ρ_{conflict}, the fraction of probes with a negative cosine. Mean cosines are near zero for every method, and the conflict rate does not follow PDR (AdamW-GradNorm has the lowest rate, 0.356, and a PDR of 0.222, while ML-FOP-SOAP has the highest, 0.711, and a PDR of 0.283).

Table S4: Raw I2T–T2I gradient-conflict statistics over the 6–8B-token window, with mean cosine (mean $\pm$ std over seeds) and conflict rate (median [Q1, Q3]).

Method	Raw C_{grad}	ρ_{conflict}
AdamW	-0.005 ± 0.009	$0.600\,[0.520, 0.678]$
Muon	-0.006 ± 0.009	$0.629\,[0.538, 0.637]$
AdamW-GradNorm	0.010 ± 0.010	$0.356\,[0.318, 0.399]$
AdamW-CAGrad	0.004 ± 0.010	$0.410\,[0.360, 0.461]$
AdamW-MGDA	-0.007 ± 0.028	$0.547\,[0.420, 0.674]$
SOAP	-0.012 ± 0.025	$0.517\,[0.469, 0.678]$
SOAP-CAGrad	0.007 ± 0.011	$0.461\,[0.383, 0.484]$
SOAP-MGDA	-0.004 ± 0.015	$0.472\,[0.386, 0.559]$
ML-FOP-SOAP	-0.009 ± 0.012	$0.711\,[0.546, 0.755]$

D Closed-Form Derivation and Numerical Guards

D.1 Local Quadratic Surrogate

Let $g_1 = g_{\mathrm{avg}} + \frac{1}{2} g_{\mathrm{diff}}$ and $g_2 = g_{\mathrm{avg}} - \frac{1}{2} g_{\mathrm{diff}}$ be the two gradients joined at a fold, $g(\beta) = g_{\mathrm{avg}} + \beta g_\perp$ the corrected candidate, and H_t the symmetric positive-definite map of Section 3.3, held fixed within the step, and inner products and norms without a subscript are taken in H_t. The surrogate is

$$J_i(\beta) = \Phi_i\big(g(\beta)\big), \qquad \Phi_i(g) = -\langle g_i, g\rangle + \tfrac{1}{2}\|g\|^2. \tag{14}$$

Its linear term is the first-order change of ℓ_i under the step $-H_t g$, and its quadratic term models the curvature of both losses by H_t^{-1}, under which the SOAP step is the Newton step of the averaged objective. This fixes the normalization. With an explicit learning rate η the linear terms scale by η and the quadratic term by η^2, both branches acquire the common term $(\eta^2 - \eta) q\beta$ with $q = \langle g_{\mathrm{avg}}, g_\perp\rangle$ of Eq. (16), the equalizing point is unchanged, and the admissible interval becomes $[-a/(2\eta b), a/(2\eta b)]$ shifted by $(1-\eta)q/(\eta b)$, a shift that vanishes under exact projection or at

$\eta = 1$. All reported coefficients use $\eta = 1$, so the interval $[-\frac{1}{2}, \frac{1}{2}]$ is a property of this normalization.

The projection is

$$c = \frac{g_{\text{diff}}^\top H_t g_{\text{avg}}}{g_{\text{avg}}^\top H_t g_{\text{avg}} + \epsilon}, \qquad g_\perp = g_{\text{diff}} - c g_{\text{avg}}. \tag{15}$$

With $A = g_{\text{diff}}^\top H_t g_{\text{avg}}$ and $D = g_{\text{avg}}^\top H_t g_{\text{avg}}$,

$$q := g_{\text{avg}}^\top H_t g_\perp = A - \frac{A}{D+\epsilon} D = \frac{\epsilon A}{D+\epsilon}, \tag{16}$$

Expanding Eq. (14) along the line,

$$J_i(\beta) = J_i(0) - \beta\, g_i^\top H_t g_\perp + \beta\, q + \frac{\beta^2}{2}\, g_\perp^\top H_t g_\perp, \tag{17}$$

and since $g_{1,2}^\top H_t g_\perp = q \pm \frac{1}{2} g_{\text{diff}}^\top H_t g_\perp$, the residual q from the linear term cancels the cross term of the quadratic term for every ϵ, and with a and b of Eq. (6) this is exactly Eq. (5), and the averaged surrogate

$$J_{\text{avg}}(\beta) = \tfrac{1}{2}\big[J_1(\beta) + J_2(\beta)\big] = J_{\text{avg}}(0) + \frac{\beta^2}{2}\, b \tag{18}$$

has no first-order term. Rearranging Eq. (15), $g_{\text{diff}} = g_\perp + c g_{\text{avg}}$, so

$$a = (g_\perp + c g_{\text{avg}})^\top H_t g_\perp = b + c\, q = b + \frac{\epsilon A^2}{(D+\epsilon)^2} \;\geq\; b. \tag{19}$$

which is the only place the stabilizer enters, so Eq. (8) is the exact minimizer for every ϵ, and its interval tends to $[-\frac{1}{2}, \frac{1}{2}]$ as $\epsilon \to 0$. Finally, since $J_i(0) = -\langle g_i, g_{\text{avg}} \rangle + \frac{1}{2} D$,

$$\Delta = J_2(0) - J_1(0) = \langle g_{\text{diff}}, g_{\text{avg}} \rangle = \tfrac{1}{2}\big(\|g_1\|^2 - \|g_2\|^2\big), \qquad \langle g_i, g_{\text{avg}} \rangle = \tfrac{1}{2}\big(\|g_i\|^2 + \langle g_1, g_2 \rangle\big). \tag{20}$$

so the uncorrected step serves each side in proportion to its squared H_t-norm, $\Delta = 0$ is equality of the two norms, and after a correction the gap is $\Delta + a\beta$.

D.2 From Total-Loss FOP to Minimax

The three rules differ in their objective, namely

$$\begin{aligned} \text{total-loss criterion (FOP)} \quad & \min_\beta \; J_{\text{tot}}(\beta), \\ \text{weak-side vertex (WSV)} \quad & \min_\beta \; J_{\text{weak}}(\beta), \\ \text{ours} \quad & \min_\beta \; \max\{J_1(\beta), J_2(\beta)\}. \end{aligned} \tag{21}$$

FOP (Lu & Armour, 2026) minimizes the total-loss model along the residual line. It projects under the Fisher metric but steps with F^{-1}, so its residual is not orthogonal to the step and the total-loss vertex is non-zero. Here one operator H_t defines both, so by Eq. (18) J_{tot} is minimized at $\beta = 0$ and cannot allocate correction between the sides. WSV minimizes the currently worse branch $w = \arg\max_i J_i(0)$, and with $s_1 = -1$ and $s_2 = +1$,

$$\beta_{\text{WSV},w} = -\frac{s_w a}{2b}, \tag{22}$$

the minimax saturation vertex on the weak side ($\pm\frac{1}{2}$ under exact projection). Minimax needs no weak-side designation, since the maximum selects the larger branch and keeps the other as a constraint.

D.3 Closed-Form Minimax Solution

Subtracting and adding the two branches of Eq. (5),

$$J_2(\beta) - J_1(\beta) = \Delta + a\beta, \tag{23}$$

and $J_1 + J_2 = 2C + b\beta^2$ with $C = \frac{1}{2}\big(J_1(0) + J_2(0)\big)$, so

$$\max\{J_1(\beta), J_2(\beta)\} = C + \frac{b}{2}\beta^2 + \frac{1}{2}|\Delta + a\beta|, \tag{24}$$

For $b > 0$ (then $a \geq b > 0$) the derivative away from the kink is $b\beta + \frac{a}{2}\operatorname{sign}(\Delta + a\beta)$ and the subdifferential at the kink $\beta_{\text{eq}} = -\Delta/a$ is $[\, b\beta_{\text{eq}} - \frac{a}{2},\ b\beta_{\text{eq}} + \frac{a}{2}\,]$, so the kink is optimal exactly when $|\beta_{\text{eq}}| \leq a/(2b)$, otherwise the optimum is the nearer boundary $\pm a/(2b)$. This is Eq. (8). The implementation returns $\hat{\beta} = 0$ when a, b or D falls below its threshold, computes the vertex as $|a|/(2b)$ with a small stabilizer in the denominator, and applies the absolute cap β_{abs} afterwards (Table S8), and regime labels in the audit exclude degenerate windows.

D.4 Relation to Preconditioned Multiple-Gradient Descent

Throughout, $\epsilon = 0$, so $\langle g_{\text{avg}}, g_\perp \rangle = 0$ and $a = b$, and $D = \|g_{\text{avg}}\|^2$. With Φ_i of Eq. (14) and $g(\beta) = g_{\text{avg}} + \beta g_\perp$,

$$\Phi_i\big(g(\beta)\big) = \Big[-\langle g_i, g_{\text{avg}}\rangle + \tfrac{1}{2}D\Big] - \beta\langle g_i, g_\perp\rangle + \tfrac{\beta^2}{2}b. \tag{25}$$

and $\langle g_1, g_\perp\rangle = \frac{1}{2}a$, $\langle g_2, g_\perp\rangle = -\frac{1}{2}a$, so this is Eq. (5) with $J_2(0) - J_1(0) = \Delta$. Over all g, $\max_i \Phi_i(g) = \max_{w\in\mathcal{S}}\big(-\langle g_w, g\rangle + \frac{1}{2}\|g\|^2\big)$ with $g_w = w_1 g_1 + w_2 g_2$ is convex in g and linear in w, so exchanging min and max gives $\max_w(-\frac{1}{2}\|g_w\|^2)$, attained at the minimum-norm point $g^\star$ of Eq. (9) (Sener & Koltun, 2018). Since $g_{\text{diff}} = g_\perp + c g_{\text{avg}}$ with $c = \Delta/D$, the segment $\operatorname{conv}\{g_1, g_2\}$ is $\{g_{\text{avg}} + \nu g_{\text{diff}} : |\nu| \leq \frac{1}{2}\}$ and $\|g_{\text{diff}}\|^2 = b + \Delta^2/D$.

(i) and (ii). If $\Delta = 0$ then $\hat{\beta} = 0$, $c = 0$ and g_{avg} is the minimum-norm point of the segment. In general the stationarity condition $\langle g_{\text{avg}} + \nu g_{\text{diff}}, g_{\text{diff}}\rangle = 0$ gives

$$\nu^\star = -\frac{\Delta}{b + \Delta^2/D} = -\kappa\,\frac{\Delta}{b}, \qquad \kappa = \frac{bD}{bD + \Delta^2}. \tag{26}$$

If $|\Delta| < b/2$ then $|\nu^\star| < \kappa/2 \leq \frac{1}{2}$ is interior and

$$g^\star = g_{\text{avg}} + \nu^\star(g_\perp + c g_{\text{avg}}) = (1 + c\nu^\star)g_{\text{avg}} + \nu^\star g_\perp = \kappa\Big(g_{\text{avg}} - \frac{\Delta}{b}\, g_\perp\Big), \tag{27}$$

using $1 + c\nu^\star = \kappa$, and Eq. (8) with $a = b$ gives $\hat{\beta} = -\Delta/b$, so $\hat{g} = g^\star/\kappa$.

(iii). $\langle \hat{g}, g_{\text{avg}}\rangle = D$ for every $\hat{\beta}$. Since g_{avg} lies in the segment, $\|g^\star\| \leq \|g_{\text{avg}}\|$, and Cauchy–Schwarz gives $\langle g^\star, g_{\text{avg}}\rangle \leq D$ with equality only if $g^\star = g_{\text{avg}}$, and in the active regime the value is κD. Property (iii) concerns step length rather than surrogate value, since on $\max_i \Phi_i$ itself the unconstrained point $g^\star$ is by definition never worse than $\hat{g}$, which is the same direction with its component along g_{avg} held at the SOAP value.

(iv). If $|\Delta| \geq b/2$ then $\hat{\beta} = -\frac{1}{2}\operatorname{sign}(\Delta)$, and the components of $\hat{g}$ along the orthogonal pair $(g_{\text{avg}}, g_\perp)$ have norms $\sqrt{D}$ and $\frac{1}{2}\sqrt{b}$, so $\tan\theta = \frac{1}{2}\sqrt{b/D}$, whereas $|\nu^\star| \geq \kappa/2$ may reach an endpoint, a single task gradient. For $|\Delta| > b/2$ the two points differ in direction, because $g^\star$ either keeps the component ratio $-\Delta/b$ of the active-regime expression, whose magnitude then exceeds $\frac{1}{2}$, or reaches an endpoint, whose direction equals that of $\hat{g}$ only if $c = 0$. At $|\Delta| = b/2$ the two directions coincide, as (ii) extends to the boundary by continuity.

Every $\hat{g}$ is $w_1 g_1 + w_2 g_2$ with $w_{1,2} = \frac{1}{2}(1 - \hat{\beta}c) \pm \hat{\beta}$, where $\hat{\beta}$ and c have opposite signs, so $w_1 + w_2 \geq 1$ with equality only at $\hat{\beta} = 0$, both weights are non-negative ($1/(2\kappa) \mp \Delta/b$ in the active regime, $|c|/4$ and $1 + |c|/4$ in the saturated one), and $\kappa(w_1, w_2)$ are the MGDA weights in the active regime. The correction is thus a per-fold conic reweighting of the two gradients, not of the losses, since the weights change at every fold and, under state isolation, never enter the SOAP statistics.

D.5 Closed-Form Audit and Comparison with a Fixed Rule

On 252 informative accumulation windows, Eq. (8) agrees with an exhaustive 20,001-point grid search over the same interval and surrogate to a mean absolute difference of 1.06×10^{-5}, every

window within $0.01\,a_t/(2b_t)$. Let $r_t = 2b_t|\Delta_t|/(a_t^2+\epsilon)$ measure local imbalance ($r_t < 1$ admits equalization, $r_t \geq 1$ is saturated) and $B_t^{(m)}$ the worse-side surrogate benefit of rule m over SOAP. Figure S1 shows the per-bin medians, where minimax is non-negative in every bin, whereas WSV is negative under light-to-moderate imbalance (-40×10^6 against $+10\times10^6$ near $r_t \approx 0.5$), as the surrogate predicts, since for $r_t < 1$, $B_t^{(\mathrm{mm})} = \frac{a_t^2}{8b_t}\, r_t(2-r_t) \geq 0$ while $B_t^{(\mathrm{WSV})} = \frac{a_t^2}{2b_t}\left(r_t - \frac{3}{4}\right)$ is negative for $r_t < \frac{3}{4}$, because the fixed step $a_t/(2b_t)$ overshoots the equalizing point $r_t\, a_t/(2b_t)$, while for $r_t \geq 1$ the two rules coincide.

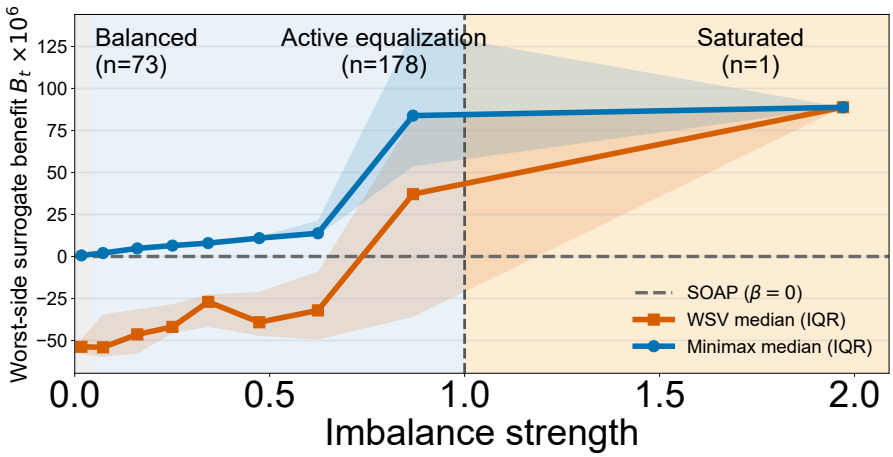

Figure S1: Worse-side surrogate benefit over SOAP on 252 training windows, per imbalance bin (median and interquartile range), for the minimax rule and the fixed weak-side-vertex rule. The vertical line marks saturation, and benefits are in unnormalized surrogate units.

E Full and Additional Results

Table S5 reports throughput and memory per architecture, Table S6 calibrates the single-run transfer gaps against the Janus-400M seed spread, Table S7 reports the update-service diagnostics of the transfer runs, Table S8 the absolute-cap sensitivity, and Figure S2 the aggregate-loss trajectories discussed in Section 4.2.

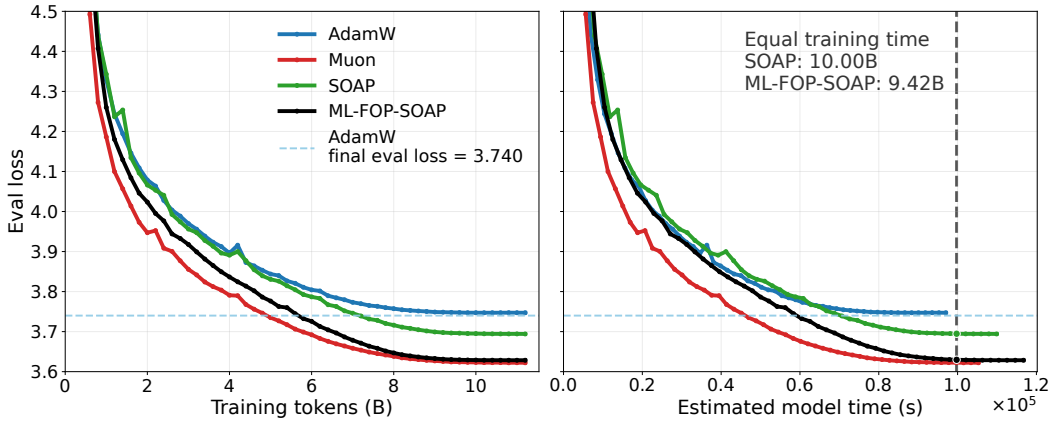

Figure S2: Aggregate validation loss of Janus-400M seed 0 against training tokens (left) and estimated model time (right). The vertical line marks equal model time, SOAP at 10.00B tokens and ML-FOP-SOAP at 9.42B, and the horizontal line is the final AdamW loss.

Table S5: Throughput and peak GPU memory per architecture. Janus-400M values are means over the main comparison, and the others are single runs. Compare within blocks only.

Janus-400M main comparison (2 GPUs, total batch 512)

Metric	AdamW	Muon	AdamW GradNorm	AdamW CAGrad	AdamW MGDA	SOAP	SOAP CAGrad	SOAP MGDA	FOP-SOAP single-shot	ML-FOP SOAP
Tokens/s ↑	101,884	96,153	95,628	104,604	104,937	100,250	95,205	89,606	96,023	94,391
Peak memory (GB) ↓	85.289	87.897	86.816	86.842	86.842	89.743	91.295	91.295	91.295	91.695

Janus-Pro-1B-official (8 GPUs, total batch 512)

Metric	AdamW	Muon	SOAP	FOP-SOAP single-shot	ML-FOP SOAP
Tokens/s ↑	84,704	79,571	82,135	76,146	74,558
Peak memory (GB) ↓	61.868	63.430	66.417	72.781	76.638

Emu3-800M (4 GPUs, total batch 512)

Metric	AdamW	Muon	SOAP	FOP-SOAP single-shot	ML-FOP SOAP
Tokens/s ↑	41,512	41,471	39,844	39,461	38,812
Peak memory (GB) ↓	145.899	146.745	148.409	149.694	149.823

Table S6: Gap over SOAP across settings. For Janus-400M the entry is the three-seed mean difference (Welch t), and for Emu3-800M and Janus-1B the single-run difference, in parentheses in units of the Janus-400M run-to-run spread $\sigma = \sqrt{s^2_{\mathrm{ML}} + s^2_{\mathrm{SOAP}}}$ (CIDEr on Emu3-800M lies in a lower range than the values behind σ). Positive favours ML-FOP-SOAP. The last column gives the share of the gain beyond the single-shot correction (400M / Emu3 / 1B), and a share above 100% means the single-shot value is below SOAP's.

Metric	Janus-400M Δ (t)	Emu3-800M Δ (Δ/σ)	Janus-1B Δ (Δ/σ)	Folding share 400M / Emu3 / 1B
I2T loss ↓	+0.025 (3.5)	+0.010 (0.8)	+0.015 (1.2)	32% / 70% / 60%
T2I loss ↓	+0.091 (7.7)	+0.041 (2.0)	+0.117 (5.7)	42% / 76% / 63%
CLIP ↑	+0.003 (0.6)	+0.007 (0.8)	+0.011 (1.2)	33% / 29% / 118%
FID-1K ↓	+4.5 (1.1)	+11.3 (1.5)	+20.9 (2.9)	37% / 47% / 70%
CIDEr-first ↑	+0.001 (0.5)	+0.024 (6.7)	+0.009 (2.5)	100% / 38% / 33%

F TASK-WEIGHT TRADE-OFFS

We vary the sampling ratio $\pi_{\mathrm{I2T}} \in \{0.25, 0.5, 0.75\}$ for AdamW, Muon, SOAP and ML-FOP-SOAP with everything else fixed, and with token-mean micro-batch losses, π_{I2T} equals an explicit loss weight in expectation. Table S9 gives the losses and downstream metrics, and Figure 2 and Section 4.2 read the loss panel as a frontier.

Table S7: Update-service diagnostics of the transfer runs (one run per row), computed as at 400M over the 60–80% stage of each token budget.

Architecture	Method	Norm ratio (parity 1)	Var. ratio (parity 1)	PDR ↑	Service bias ↓
Emu3-800M	AdamW	0.670	1.502	0.222	0.444
Emu3-800M	Muon	0.578	0.711	0.316	0.281
Emu3-800M	SOAP	0.610	0.354	0.292	0.154
Emu3-800M	FOP-SOAP single-shot	0.680	0.819	0.254	0.077
Emu3-800M	ML-FOP-SOAP	0.726	0.854	0.331	0.038
Janus-1B	AdamW	0.716	0.259	0.241	0.076
Janus-1B	Muon	1.185	0.352	0.276	0.055
Janus-1B	SOAP	1.041	0.527	0.253	0.010
Janus-1B	FOP-SOAP single-shot	1.038	0.626	0.274	0.015
Janus-1B	ML-FOP-SOAP	1.062	0.728	0.291	0.002

Table S8: Absolute-cap sensitivity of ML-FOP-SOAP on Janus-400M, where the reported runs use $\beta_{\text{abs}} = 0.50$ and SOAP corresponds to $\beta = 0$.

Method	Absolute cap	Eval I2T ↓	Eval T2I ↓	PDR ↑	Service bias ↓
SOAP	0	0.697	6.671	0.262	0.071
ML-FOP-SOAP	0.25	0.684	6.616	0.273	0.040
ML-FOP-SOAP	0.50	0.672	6.580	0.283	0.009
ML-FOP-SOAP	1.00	0.676	6.578	0.281	0.009

Table S9: Task-sampling ratio sweep on Janus-400M ("Mix" is I2T:T2I). All rows are mean ± standard deviation over three seeds. (a) validation losses, (b) downstream metrics.

(a) Training configuration and validation losses

Method	Mix	Token budget	Eval loss ↓	Eval I2T ↓	Eval T2I ↓
AdamW	25:75	10B	3.744 ± 0.004	0.891 ± 0.005	6.597 ± 0.009
AdamW	50:50	10B	3.740 ± 0.008	0.741 ± 0.004	6.738 ± 0.017
AdamW	75:25	10B	3.811 ± 0.005	0.674 ± 0.005	6.948 ± 0.006
Muon	25:75	10B	3.682 ± 0.007	0.792 ± 0.005	6.573 ± 0.013
Muon	50:50	10B	3.622 ± 0.011	0.568 ± 0.001	6.675 ± 0.023
Muon	75:25	10B	3.745 ± 0.009	0.593 ± 0.007	6.897 ± 0.009
SOAP	25:75	10B	3.695 ± 0.006	0.817 ± 0.005	6.573 ± 0.009
SOAP	50:50	10B	3.684 ± 0.009	0.697 ± 0.007	6.671 ± 0.010
SOAP	75:25	10B	3.753 ± 0.008	0.636 ± 0.005	6.869 ± 0.014
ML-FOP-SOAP	25:75	10B	3.638 ± 0.004	0.815 ± 0.004	6.459 ± 0.004
ML-FOP-SOAP	50:50	10B	3.625 ± 0.012	0.672 ± 0.010	6.580 ± 0.018
ML-FOP-SOAP	75:25	10B	3.655 ± 0.009	0.613 ± 0.002	6.697 ± 0.016

(b) Downstream metrics for the same rows

Method	Mix	Gen. CLIP ↑	FID-1K ↓	MMKC-VS ↑	CIDEr-first ↑	BLEU-first ↑
AdamW	25:75	0.165 ± 0.004	187.134 ± 8.958	0.306 ± 0.013	0.370 ± 0.005	0.113 ± 0.005
AdamW	50:50	0.172 ± 0.006	191.117 ± 8.182	0.265 ± 0.012	0.377 ± 0.006	0.115 ± 0.002
AdamW	75:25	0.152 ± 0.008	172.689 ± 9.441	0.299 ± 0.012	0.391 ± 0.008	0.119 ± 0.004
Muon	25:75	0.167 ± 0.007	245.012 ± 4.699	0.282 ± 0.015	0.377 ± 0.001	0.115 ± 0.003
Muon	50:50	0.165 ± 0.009	190.520 ± 5.023	0.234 ± 0.014	0.384 ± 0.002	0.117 ± 0.003
Muon	75:25	0.169 ± 0.011	194.468 ± 4.352	0.288 ± 0.011	0.407 ± 0.003	0.128 ± 0.004
SOAP	25:75	0.152 ± 0.005	200.086 ± 7.324	0.286 ± 0.024	0.371 ± 0.002	0.115 ± 0.005
SOAP	50:50	0.174 ± 0.006	179.369 ± 6.923	0.240 ± 0.031	0.380 ± 0.003	0.118 ± 0.002
SOAP	75:25	0.169 ± 0.008	213.641 ± 8.487	0.245 ± 0.029	0.380 ± 0.004	0.117 ± 0.002
ML-FOP-SOAP	25:75	0.180 ± 0.005	178.827 ± 2.776	0.331 ± 0.023	0.371 ± 0.003	0.118 ± 0.004
ML-FOP-SOAP	50:50	0.177 ± 0.007	174.838 ± 2.287	0.286 ± 0.024	0.381 ± 0.002	0.120 ± 0.003
ML-FOP-SOAP	75:25	0.174 ± 0.009	164.715 ± 1.929	0.308 ± 0.018	0.401 ± 0.001	0.125 ± 0.002

G PSEUDOCODE

Algorithm 1 gives one ML-FOP-SOAP step for the reported configuration ($K = 4$), and Section 3.4 describes the general recursion.

Algorithm 1 ML-FOP-SOAP step for one accumulation window ($K = 4$, dyadic checkpoints $1, 2, 4$, while the general case folds the next segment at every checkpoint).

Input micro-step gradients $g^{(1)}, \dots, g^{(4)}$, each from a direction-pure micro-batch, and the SOAP state defining H_t, held fixed for this step
$\bar{g} \leftarrow \frac{1}{4} \sum_k g^{(k)}$, then update the SOAP curvature, eigenbasis and second-moment statistics from $\bar{g}$ ▷ state isolation
$z \leftarrow g^{(1)}$
for segment s in $\big(g^{(2)},\ (g^{(3)} + g^{(4)})/2\big)$ **do**
  $g_{\text{avg}} \leftarrow \frac{1}{2}(z + s), \quad g_{\text{diff}} \leftarrow z - s$
  $g_\perp \leftarrow g_{\text{diff}} - c\, g_{\text{avg}}$ with c from Eq. (2) ▷ residual orthogonal to the SOAP step
  a, b, Δ from Eq. (6) and $\hat{\beta} \leftarrow \text{clip}(-\Delta/a,\ -a/(2b),\ a/(2b))$ ▷ Eq. (8), guards in Appendix B
  $z \leftarrow g_{\text{avg}} + \hat{\beta}\, g_\perp$ ▷ corrected running aggregate
end for
Output update direction $P_t(z)$ ▷ equals $P_t(\bar{g})$ when every $\hat{\beta} = 0$