Appendix A. Experimental Details

All code is available at `github.com/jxbz/nero`. This appendix records important details of the implementations.

MNIST classification These experiments used a multilayer perceptron (MLP) network. An L-layer architecture consisted of $(L-1)$ layers of dimension 784×784 followed by an output layer of dimension 784×10. A "scaled relu" nonlinearity was used, defined by $\varphi(x) := \sqrt{2} \cdot \max(0, x)$. The factor of $\sqrt{2}$ was motivated by *Kaiming init* (He et al., 2015) and was not tuned. The reparameterisation experiment used $L = 5$ layers and trained for 5 epochs without learning rate decay. The very deep MLP used $L = 100$ layers and trained for 50 epochs with the learning rate decayed by a factor of 0.9 at the end of every epoch, and with the initial learning rate tuned over $\{0.0001, 0.001, 0.01, 0.1\}$. Training took place on an unknown Google Colab GPU. On an NVIDIA Tesla P100 GPU, the 5-layer MLP took ~ 1 minute to train and the 100-layer MLP took ~ 30 minutes.

CIFAR-10 cGAN Equal learning rates were used in the generator and discriminator. The initial learning rate was tuned over {0.0001, 0.001, 0.01, 0.1, 1.0} for all optimisers. The networks were trained for 120 epochs, with the learning rate decayed by a factor of 10 at epoch 100. The momentum parameter in SGD and β_1 in Adam and LAMB were tuned over {0.0, 0.9}. Nero's β and β_2 in Adam and LAMB were set to 0.999 without tuning. Training took around 3 hours on an NVIDIA RTX 2080Ti GPU.

CIFAR-10 classification All models were trained for 200 epochs, with 5 epochs of linear learning rate warm-up and learning rate decay by a factor of 0.2 at epochs 100, 150 and 180. The initial learning rates were tuned over {0.0001, 0.001, 0.01, 0.1, 1.0}. Training was performed on an NVIDIA RTX 2080Ti GPU. Training time for the VGG-11 network was ~ 1 hour, and for ResNet-18 was ~ 2 hours.

Since the experiments in Figures 1 and 2 were intended to probe the fundamental properties of optimisers rather than their performance under a limited tuning budget, a more fine-grained learning rate search was conducted. Specifically, the learning rates were tuned over {0.01, 0.02, 0.04, 0.06, 0.08, 0.1}. The best results are listed in the following table:

Optimiser	Fix Mean	Fix Norm	Top-1 Error	Best η
Nero			$12.17\% \pm 0.08$	0.02
Nero	✓		$11.99\% \pm 0.14$	0.01
Nero		✓	$10.03\% \pm 0.24$	0.02
Nero	✓	✓	$\mathbf{8.61\% \pm 0.22}$	0.02
Madam			$12.77\% \pm 0.20$	0.02
Madam	✓	✓	$12.60\% \pm 0.12$	0.06
LAMB			$12.73\% \pm 0.10$	0.02
LAMB	✓	✓	$8.88\% \pm 0.08$	0.06

ImageNet classification For training with SGD + momentum + weight decay, the initial learning was set to 0.1, momentum was set to 0.9 and weight decay was set to 0.0001. These settings follow He et al. (2016). One epoch of linear learning rate warm-up was used, followed by 89 epochs of cosine annealing. The batch size was set to 400 for ResNet-50 to fit the GPU vRAM budget, and this was in the range known to yield good performance (Goyal et al., 2017). This paper's SGD implementation surpassed the target ImageNet top-1 accuracy of 76.3% for ResNet-50 (Goyal et al., 2017; You et al., 2020). The training was distributed over four NVIDIA RTX 2080Ti GPUs, taking ~ 45 hours per training run. The total number of GPU hours for all ImageNet experiments in this paper was ~ 1500.

Wikitext-2 language model The small transformer model was trained for 20 epochs, with the learning rate decayed by a factor of 0.1 at epoch 10. The initial learning rate was tuned over {0.0001, 0.001, 0.01, 0.1, 1.0}. The batch size was set to 20. Training on an NVIDIA RTX 2080Ti GPU took ~ 15 minutes.

WMT16 En–De translation The large transformer model was trained for 100 epochs, with a linear warm-up from epoch 0 to 50, and linear annealing from epoch 50 to 100. The maximum learning rate was tuned over {0.0001, 0.001, 0.01, 0.1, 1.0}. A batch size of 128 was used. Training took ~ 1 hour on an NVIDIA RTX 2080Ti GPU.

Reinforcement learning Hyperparameter settings followed Kostrikov (2018), except for the initial learning rate and the total number of environment steps. The number of steps was fixed to 5 million, and the initial learning rate was tuned over {0.0001, 0.001, 0.01, 0.1, 1.0}. The policy network combined convolutional image feature extractors with dense output layers. Training was performed on an NVIDIA RTX 2080Ti GPU, and the training time was ~ 1.5 hours.