Supplementary material

N. Maheswaranathan, L. Metz, G. Tucker, D. Choi and J. Sohl-Dickstein

A. Derivation of the bias and variance of the Guided ES update

A.1. Bias

We can evaluate the squared norm of the bias of our estimate g as

$$\begin{aligned}\|\operatorname{Bias}(g)\|_2^2 &= (\mathbb{E}[g] - \nabla f(x))^T (\mathbb{E}[g] - \nabla f(x)) \\ &= \nabla f(x)^T \left(\beta\Sigma - I\right)^2 \nabla f(x).\end{aligned}$$

We additionally define the *normalized* squared bias, $\tilde{b}$, as the squared norm of the bias divided by the squared norm of the true gradient (this quantity is independent of the overall scale of the gradient).

$$\|\mathbb{E}[g] - \nabla f(x)\|_2^2 = \nabla f(x)^T \left(\beta\Sigma - I\right)^2 \nabla f(x),$$

where $\epsilon \sim \mathcal{N}(0, \Sigma)$ and the covariance is given by: $\Sigma = \frac{\alpha}{n} I + \frac{1-\alpha}{k} U U^T$. This expression reduces to (recall that U is orthonormal, so $U^T U = I$):

$$\begin{aligned}\|\operatorname{Bias}\|_2^2 &= \|\nabla f(x)\|_2^2 \left[\beta^2 \frac{\alpha^2}{n^2} - 2\beta\frac{\alpha}{n} + 1 + \left(\beta^2 \frac{(1-\alpha)^2}{k^2} + 2\beta^2 \frac{\alpha(1-\alpha)}{kn} - 2\beta \frac{1-\alpha}{k}\right) \|\rho\|_2^2\right] \\ &= \|\nabla f(x)\|_2^2 \left[\left(\beta\frac{\alpha}{n} - 1\right)^2 + \left(\beta^2 \frac{(1-\alpha)^2}{k^2} + 2\beta \frac{1-\alpha}{k}\left(\beta\frac{\alpha}{n} - 1\right)\right) \|\rho\|_2^2\right]\end{aligned}$$

$$\tilde{b} = \left(\beta\frac{\alpha}{n} - 1\right)^2 + \left(\beta^2 \frac{(1-\alpha)^2}{k^2} + 2\beta \frac{(1-\alpha)}{k}\left(\beta\frac{\alpha}{n} - 1\right)\right) \|\rho\|_2^2 \tag{6}$$

where again β is a scale factor and α is part of the parameterization of the covariance matrix that trades off variance in the full parameter space for variance in the guiding subspace ($\Sigma = \frac{\alpha}{n} I + \frac{(1-\alpha)}{k} U U^T$). We see that the normalized squared bias consists of two terms: the first is a contribution from the search in the full space and is thus independent of ρ, whereas the second depends on the squared norm of the uncentered correlation, $\|\rho\|_2^2$.

A.2. Variance

In addition to the bias, we are also interested in the variance of our estimate. First, we state a useful identity. Suppose $\epsilon \sim \mathcal{N}(0, \Sigma)$, then

$$\mathbb{E}[\epsilon\epsilon^T \epsilon\epsilon^T] = \operatorname{tr}(\Sigma)\Sigma + 2\Sigma^2.$$

We can see this by observing that the (i, k) entry of $\mathbb{E}[\epsilon\epsilon^T \epsilon\epsilon^T] = \mathbb{E}[(\epsilon^T\epsilon)\epsilon\epsilon^T]$ is

$$\begin{aligned}\mathbb{E}\left[\sum_j \epsilon_i \epsilon_j^2 \epsilon_k\right] &= \sum_j \mathbb{E}\left[\epsilon_i \epsilon_j^2 \epsilon_k\right] \\ &= \sum_j \mathbb{E}\left[\epsilon_j^2\right] \mathbb{E}\left[\epsilon_i \epsilon_k\right] + 2 \sum_j \mathbb{E}\left[\epsilon_i \epsilon_j\right] \mathbb{E}\left[\epsilon_j \epsilon_k\right],\end{aligned}$$

by Isserlis' theorem, and then we recover the identity by rewriting the terms in matrix notation.

We use total variance (i.e., $\operatorname{tr}(\operatorname{Var}(g))$) to quantify the variance of our estimator:

$$\begin{aligned}\text{total variance} \equiv \operatorname{tr}\left(\operatorname{Var}(g)\right) &= \operatorname{tr}\left(\mathbb{E}[gg^T] - \mathbb{E}[g]\mathbb{E}[g]^T\right) = \mathbb{E}[g^T g] - \mathbb{E}[g]^T \mathbb{E}[g] \\ &= \beta^2 \nabla f(x)^T \mathbb{E}[\epsilon\epsilon^T \epsilon\epsilon^T] \nabla f(x) - \beta^2 \nabla f(x)^T \Sigma^T \Sigma \nabla f(x)\end{aligned} \tag{7}$$

Using the identity above, we can express the total variance as:

$$\begin{aligned}\text{total variance} &= \beta^2 \nabla f(x)^T \left(\text{tr}(\Sigma)\Sigma + 2\Sigma^2\right) \nabla f(x) - \beta^2 \nabla f(x)^T \Sigma^2 \nabla f(x) \\ &= \beta^2 \nabla f(x)^T \left(\text{tr}(\Sigma)\Sigma + \Sigma^2\right) \nabla f(x)\end{aligned}$$

Since the trace of the covariance matrix Σ is 1, we can expand the quantity $\text{tr}(\Sigma)\Sigma + \Sigma^2$ as:

$$\begin{aligned}\text{tr}(\Sigma)\Sigma + \Sigma^2 &= \Sigma + \Sigma^2 \\ &= \left[\frac{\alpha^2}{n^2} + \frac{\alpha}{n}\right] I + \left[\frac{(1-\alpha)^2}{k^2} + 2\frac{\alpha(1-\alpha)}{kn} + \frac{1-\alpha}{k}\right] UU^T\end{aligned}$$

Thus the expression for the total variance reduces to:

$$\text{total variance} = \|\nabla f(x)\|_2^2 \beta^2 \left(\frac{\alpha^2}{n^2} + \frac{\alpha}{n} + \left[\frac{(1-\alpha)^2}{k^2} + 2\frac{\alpha(1-\alpha)}{kn} + \frac{1-\alpha}{k}\right] \|\rho\|_2^2\right),$$

and dividing by the norm of the gradient yields the expression for the normalized variance (eq. (4) in the main text).

B. Optimal hyperparameters

B.1. Reparameterization

We wish to minimize the sum of the normalized bias and variance, eq. (5) in the main text. First, we use a reparameterization by using the substitution $\theta_1 = \alpha\beta$ and $\theta_2 = (1-\alpha)\beta$. This substitution yields:

$$\tilde{b} + \tilde{v} = \left[2\frac{\theta_1^2}{n^2} + (\theta_0 + \theta_1 - 2)\frac{\theta_0}{n} + 1\right] + \left[2\frac{\theta_2^2}{k^2} + 4\frac{\theta_0\theta_1}{kn} + (\theta_0 + \theta_1 - 2)\frac{\theta_1}{k}\right] \|\rho\|_2^2,$$

which is quadratic in θ. Therefore, we can rewrite the problem as: $\tilde{b} + \tilde{v} = \frac{1}{2}\|A\theta - b\|_2^2$, where A and b are given by:

$$A = \begin{pmatrix} \frac{2}{n^2} + \frac{1}{n} & \frac{1}{2}\left(\frac{4\|\rho\|_2^2}{kn} + \frac{\|\rho\|_2^2}{k} + \frac{1}{n}\right) \\ \frac{1}{2}\left(\frac{4\|\rho\|_2^2}{kn} + \frac{\|\rho\|_2^2}{k} + \frac{1}{n}\right) & \left(\frac{2}{k^2} + \frac{1}{k}\right)\|\rho\|_2^2 \end{pmatrix}, b = \begin{pmatrix} \frac{1}{n} \\ \frac{\|\rho\|_2^2}{k} \end{pmatrix} \tag{8}$$

Note that A and b depend on the problem data (k, n, and $\|\rho\|_2$), and that A is a positive semi-definite matrix (as k and n are non-negative integers, and $\|\rho\|_2$ is between 0 and 1). In addition, we can express the constraints on the original parameters ($\beta \geq 0$ and $0 \leq \alpha \leq 1$) as a non-negativity constraint in the new parameters ($\theta \succeq 0$).

B.2. KKT conditions

The optimal hyperparameters are defined (see main text) as the solution to the minimization problem:

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \tfrac{1}{2}\|A\theta - b\|_2^2 \\ \text{subject to} \quad & \theta \succeq 0 \end{aligned} \tag{9}$$

where $\theta = \begin{pmatrix} \alpha\beta \\ (1-\alpha)\beta \end{pmatrix}$ are the hyperparameters to optimize, and A and b are specified in eq. (8).

The Lagrangian for (9) is given by $L(\theta, \lambda) = \frac{1}{2}\|A\theta - b\|_2^2 - \lambda^T\theta$, and the corresponding dual problem is:

$$\begin{aligned} \underset{\lambda}{\text{maximize}} \quad & \inf_\theta \tfrac{1}{2}\|A\theta - b\|_2^2 - \lambda^T\theta \\ \text{subject to} \quad & \lambda \succeq 0 \end{aligned} \tag{10}$$

Since the primal is convex, we have strong duality and the Karush-Kuhn-Tucker (KKT) conditions guarantee primal and dual optimality. These conditions include primal and dual feasibility, that the gradient of the Lagrangian vanishes

($\nabla_\theta L(\theta, \lambda) = A\theta - b - \lambda = 0$), and complimentary slackness (which ensures that for each inequality constraint, either the constraint is satisfied or $\lambda = 0$).

Solving the condition on the gradient of the Langrangian for λ yields that the lagrange multipliers λ are simply the residual $\lambda = A\theta - b$. Complimentary slackness tells us that $\lambda_i \theta_i = 0$, for all i. We are interested in when this constraint becomes tight. To solve for this, we note that there are two regimes where each of the two inequality constraints is tight (the blue and orange regions in Figure 3a). These occur for the solutions $\theta^{(1)} = \begin{pmatrix} 0 \\ \frac{k}{k+2} \end{pmatrix}$ (when the first inequality is tight) and $\theta^{(2)} = \begin{pmatrix} \frac{n}{n+2} \\ 0 \end{pmatrix}$ (when the second inequality is tight). To solve for the transition point, we solve for the point where the constraint is tight *and* the lagrange multiplier (λ) equals zero. We have two inequality constraints, and thus will have two solutions (which are the two solid curves in Figure 3a). Since the lagrange multiplier is the residual, these points occur when $\left(A\theta^{(1)} - b\right)_1 = \lambda_1 = 0$ and $\left(A\theta^{(2)} - b\right)_2 = \lambda_2 = 0$.

The first solution $\theta^{(1)} = \begin{pmatrix} 0 \\ \frac{k}{k+2} \end{pmatrix}$ yields the upper bound:

$$\begin{aligned} \left(A\theta^{(1)}\right)_1 - b_1 &= 0 \\ \frac{1}{2}\left(\frac{1}{n} + \frac{\|\rho\|_2^2}{k} + 4\frac{\|\rho\|_2^2}{kn}\right)\left(\frac{k}{k+2}\right) &= \frac{1}{n} \\ \|\rho\|_2^2\left(\frac{n+4}{n}\right) &= \frac{k+4}{n} \\ \|\rho\|_2 &= \sqrt{\frac{k+4}{n+4}} \end{aligned}$$

And the second solution $\theta^{(2)} = \begin{pmatrix} \frac{n}{n+2} \\ 0 \end{pmatrix}$ yields the lower bound:

$$\begin{aligned} \left(A\theta^{(2)}\right)_2 - b_2 &= 0 \\ \frac{1}{2}\left(\frac{1}{n} + \frac{\|\rho\|_2^2}{k} + 4\frac{\|\rho\|_2^2}{kn}\right)\left(\frac{n}{n+2}\right) &= \frac{\|\rho\|_2^2}{k} \\ k + n\|\rho\|_2^2 + 4\|\rho\|_2^2 &= \|\rho\|_2^2(2n+4) \\ \|\rho\|_2 &= \sqrt{\frac{k}{n}} \end{aligned}$$

These are the equations for the lines separating the regimes of optimal hyperparameters in Figure 3.

C. Alternative motivation for optimal hyperparameters

Choosing hyperparameters which most rapidly descend the simple quadratic loss in eq. (11) is equivalent to choosing hyperparameters which minimize the expected square error in the estimated gradient, as is done in §3.4. This provides further support for the method used to choose hyperparameters in the main text. Here we derive this equivalence.

Assume a loss function of the form

$$f(x) = \frac{1}{2}\|x\|_2^2, \tag{11}$$

and that updates are performed via gradient descent with learning rate 1,

$$x \leftarrow x - g.$$

The expected loss after a single training step is then

$$\mathbb{E}_g\left[f\left(x - g\right)\right] = \frac{1}{2}\mathbb{E}_g\left[\|x - g\|_2^2\right]. \tag{12}$$

For this problem, the true gradient is simply $\nabla f(x) = x$. Substituting this into eq. (12), we find

$$\mathbb{E}_g \left[f(x-g) \right] = \frac{1}{2} \mathbb{E}_g \left[\|\nabla f(x) - g\|_2^2 \right].$$

Up to a multiplicative constant, this is exactly the expected square error between the descent direction g and the gradient $\nabla f(x)$ used as the objective for choosing hyperparameters in §3.4.

D. Experimental details

Below, we give detailed methods used for each of the experiments from §4. For each problem, we specify a desired loss function that we would like to minimize ($f(x)$), as well as specify the method for generating a surrogate or approximate gradient ($\nabla \tilde{f}(x)$).

D.1. Quadratic function with a biased gradient

Our target problem is linear regression, $f(x) = \frac{1}{2M} \|Ax - b\|_2^2$, where A is a random $M \times N$ matrix and b is a random M-dimensional vector. The elements of A and b were drawn IID from a standard Normal distribution. We chose $N = 1000$ and $M = 2000$ for this problem. The surrogate gradient was generated by adding a random bias (drawn once at the beginning of optimization) and noise (resampled at every iteration) to the gradient. These quantities were scaled to have the same norm as the gradient. Thus, the surrogate gradient is given by: $\nabla \tilde{f}(x) = \nabla f(x) + (b + n) \|\nabla f(x)\|_2$, where b and n are unit norm random vectors that are fixed (bias) or resampled (noise) at every iteration.

The plots in Figure 1b show the loss suboptimality ($f(x) - f^*$), where f^* is the minimum of $f(x)$ for a particular realization of the problem. The parameters were initialized to the zeros vector and optimized for 10,000 iterations. Figure 1b shows the mean and spread (std. error) over 10 random seeds. For each optimization algorithm, we performed a coarse grid search over the learning rate for each method, scanning 17 logarithmically spaced values over the range $(10^{-5}, 1)$. The learning rates chosen were: 5e-3 for gradient descent, 0.2 for guided and vanilla ES, and 1.0 for CMA-ES. For the two evolutionary strategies algorithms, we set the overall variance of the perturbations as $\sigma = 0.1$ and used $P = 1$ pair of samples per iteration. The subspace dimension for Guided ES was set to $k = 10$. The results were not sensitive to the choices for σ, P, or k.

D.2. Unrolled optimization

We define the target problem as the loss of a quadratic after running $T = 15$ steps of gradient descent. The quadratic has the same form as described above, $\frac{1}{2M} \|Ax - b\|_2^2$, but with $M = 20$ and $N = 10$. The learning rate for the optimizer was taken as the output of a multilayer perceptron (MLP), with three hidden layers containing 32 hidden units per layer and with rectified linear (ReLU) activations after each hidden layer. The inputs to the MLP were the 10 eigenvalues of the Hessian, $A^T A$, and the output was a single scalar that was passed through a softplus nonlinearity (to ensure a positive learning rate). Note that the optimal learning rate for this problem is $\frac{2M}{\lambda_{\text{min}} + \lambda_{\text{max}}}$, where λ_{min} and λ_{max} are the minimum and maximum eigenvalues of $A^T A$, respectively.

The surrogate gradients for this problem were generated by backpropagation through the optimization process, but by unrolling only $T = 1$ optimization steps (truncated backprop). Figure 4b shows the distance between the MLP predicted learning rate and the optimal learning rate $\left(\frac{2M}{\lambda_{\text{min}} + \lambda_{\text{max}}} \right)$, during the course of optimization of the MLP parameters. That is, Figure 4b shows the progress on the meta-optimization problems (optimizing the MLP to predict the learning rate) using the three different algorithms (SGD, vanilla ES, and guided ES).

As before, the mean and spread (std. error) over 10 random seeds are shown, and the learning rate for each of the three methods was chosen by a grid search over the range $(10^{-5}, 10)$. The learning rates chosen were 0.3 for gradient descent, 0.5 for guided ES, and 10 for vanilla ES. For the two evolutionary strategies algorithms, we set the variance of the perturbations to $\sigma = 0.01$ and used $P = 1$ pair of samples per iteration. The results were not sensitive to the choices for σ, P, or k.

D.3. Synthesizing gradients for a guiding subspace

Here, the target problem consisted of a mean squared error objective, $f(x) = \frac{1}{2} \|x - x^*\|_2^2$, where x^* was random sampled from a uniform distribution between [-1, 1]. The surrogate gradient was defined as the gradient of a model, $M(x; \theta)$, with

inputs x and parameters θ. We parameterize this model using a multilayered perceptron (MLP) with two 64-unit hidden layers and relu activations. The surrogate gradients were taken as the gradients of M with respect to x: $\nabla \tilde{f}(x) = \nabla_x M(x; \theta)$.

The model was optimized online during optimization of f by minimizing the mean squared error with the (true) function observations: $L_{\text{model}}(\theta) = \mathbb{E}_{x \sim D} \left[f(x) - M(x; \theta) \right]^2$. The data D used to train M were randomly sampled in batches of size 512 from the most recent 8192 function evaluations encountered during optimization. This is equivalent to uniformly sampling from a replay buffer, a strategy commonly used in reinforcement learning. We performed one θ update per x update with Adam with a learning rate of 1e-4.

The two evolutionary strategies algorithms inherently generate samples of the function during optimization. In order to make a fair comparison when optimizing with the Adam baseline, we similarly generated function evaluations for training the model M by sampling points around the current iterate from the same distribution used in vanilla ES (Normal with $\sigma = 0.1$). This ensures that the amount and spread of training data for M (in the replay buffer) when optimizing with Adam is similar to the data in the replay buffer when training with vanilla or guided ES.

Figure 5a shows the mean and spread (standard deviation) of the performance of the three algorithms over 10 random instances of the problem. We set $\sigma = 0.1$ and used $P = 1$ pair of samples per iteration. For Guided ES, we used a subspace dimension of $k = 1$. The results were not sensitive to the number of samples P, but did vary with σ, as this controls the spread of the data used to train M, thus we tuned σ with a coarse grid search.