A. Derivation of the ODE

The derivation of the ODE's that describe the dynamics of the test error for shallow networks closely follows the one of Saad & Solla (1995a) and Biehl & Schwarze (1995) for back-propagation. Here, we give the main steps to obtain the analytical curves of the main text and refer the reader to their paper for further details.

As we discuss in Sec. 1, student and teacher are both two-layer networks with K and M hidden nodes, respectively. For an input $x \in \mathbb{R}^N$, their outputs $\hat{y}$ and y can be written as

$$\hat{y} = \phi_\theta(x) = \sum_{k=1}^{K} W_2^k \, g\left(\lambda^k\right), \qquad y = \phi_{\tilde{\theta}}(x) = \sum_{m=1}^{M} \tilde{W}_2^m \, g\left(\nu^m\right), \tag{21}$$

where we have introduced the pre-activations $\lambda^k \equiv W_1^k x/\sqrt{N}$ and $\nu^m \equiv \tilde{W}_1^m x/\sqrt{N}$. Evaluating the test error of a student with respect to the teacher under the squared loss leads us to compute the average

$$\epsilon_g\left(\theta, \tilde{\theta}\right) = \frac{1}{2}\mathbb{E}_x \left[\sum_{k=1}^{K} W_2^k \, g\left(\lambda^k\right) - \sum_{m=1}^{M} \tilde{W}_2^m \, g\left(\nu^m\right)\right]^2, \tag{22}$$

where the expectation is taken over inputs x for a fixed student and teacher. Since x only enters Eq. (22) via the pre-activations $\lambda = (\lambda^k)$ and $\nu = (\nu^m)$, we can replace the high-dimensional average over x by a low-dimensional average over the $K + M$ variables (λ, ν). The pre-activations are jointly Gaussian since the inputs are drawn element-wise i.i.d. from the Gaussian distribution. The mean of (λ, ν) is zero since $\mathbb{E}\, x_i = 0$, so the distribution of (λ, ν) is fully described by the second moments

$$Q^{kl} = \mathbb{E}\, \lambda^k \lambda^l = W_1^k \cdot W_1^l / N, \tag{23}$$

$$R^{km} = \mathbb{E}\, \lambda^k \nu^m = W_1^k \cdot \tilde{W}_1^m / N, \tag{24}$$

$$T^{mn} = \mathbb{E}\, \nu^m \nu^n = \tilde{W}_1^m \cdot \tilde{W}_1^n / N. \tag{25}$$

which are the "order parameters" that we introduced in the main text. We can thus rewrite the generalisation error (5) as a function of only the order parameters and the second-layer weights,

$$\lim_{N\to\infty} \epsilon_g(\theta, \tilde{\theta}) = \epsilon_g(Q, R, T, W_2, \tilde{W}_2) \tag{26}$$

As we update the weights using SGD, the time-dependent order parameters Q, R, and W_2 evolve in time. By choosing different scalings for the learning rates in the SGD updates (4), namely

$$\eta_{W_1} = \eta, \qquad \eta_{W_2} = \eta/N$$

for some constant η, we guarantee that the dynamics of the order parameters can be described by a set of ordinary differential equations, called their "equations of motion". We can obtain these equations in a heuristic manner by squaring the weight update (4) and taking inner products with $\tilde{W}_1^m$, to yield the equations of motion for Q and R respectively:

$$\frac{\mathrm{d}R^{km}}{\mathrm{d}\alpha} = -\eta F_1^k \mathbb{E}\left[g'(\lambda^k)\nu^m e\right] \tag{27a}$$

$$\frac{\mathrm{d}Q^{k\ell}}{\mathrm{d}\alpha} = -\eta F_1^k \mathbb{E}\left[g'(\lambda^k)\lambda^\ell e\right] - \eta F_1^\ell \mathbb{E}\left[g'(\lambda^\ell)\lambda^k e\right] + \eta^2 F_1^k F_1^\ell \mathbb{E}\left[g'(\lambda^k) g'(\lambda^\ell) e^2\right], \tag{27b}$$

$$\frac{\mathrm{d}W_2^k}{\mathrm{d}\alpha} = -\eta \mathbb{E}\left[g(\lambda^k) e\right] \tag{27c}$$

where, as in the main text, we introduced the error $e = \phi_\theta(x) - \phi_{\tilde{\theta}}(x)$. In the limit $N \to \infty$, the variable $\alpha = \mu/N$ becomes a continuous time-like variable. The remaining averages over the pre-activations, such as

$$\mathbb{E}\, g'(\lambda^k)\lambda^\ell g(\nu^m),$$

are simple three-dimensional integral over the Gaussian random variables λ^k, λ^ℓ and ν^m and can be evaluated analytically for the choice of $g(x) = \mathrm{erf}(x/\sqrt{2})$ (Biehl & Schwarze, 1995) and for linear networks with $g(x) = x$. Furthermore, these averages can be expressed only in term of the order parameters, and so the equations close. We note that the asymptotic exactness of Eqs. 27 can be proven using the techniques used recently to prove the equations of motion for BP (Goldt et al., 2019).

We provide an integrator for the full system of ODEs for any K and M in the Github repository.

B. Detailed analysis of DFA dynamics

In this section, we present a detailed analysis of the ODE dynamics in the matched case $K = M$ for sigmoidal networks ($g(x) = \mathrm{erf}\left(x/\sqrt{2}\right)$).

The Early Stages and Gradient Alignment We now use Eqs. (27) to demonstrate that alignment occurs in the early stages of learning, determining from the start the solution DFA will converge to (see Fig. 3 which summarises the dynamical evolution of the student's second layer weights).

Assuming zero initial weights for the student and orthogonal first layer weights for the teacher (i.e. T^{nm} is the identity matrix), for small times ($t \ll 1$), one can expand the order parameters in t:

$$R^{km}(t) = t\dot{R}^{km}(0) + \mathcal{O}(t^2), \qquad Q^{kl}(t) = t\dot{Q}^{kl}(0) + \mathcal{O}(t^2), \qquad W_2^k(t) = t\dot{W}_2^k(0) + \mathcal{O}(t^2). \tag{28}$$

where, due to the initial conditions, $R(0) = Q(0) = W_2(0) = 0$. Using Eq. 27, we can obtain the lowest order term of the above updates:

$$\begin{aligned}\dot{R}^{km}(0) &= \frac{\sqrt{2}}{\pi}\eta\tilde{W}_2^m F_1^k,\\ \dot{Q}^{kl}(0) &= \frac{2}{\pi}\eta^2\left((\tilde{W}_2^k)^2 + (\tilde{W}_2^l)^2\right)F_1^l F_1^k,\\ \dot{W}_2^k(0) &= 0\end{aligned} \tag{29}$$

Since both $\dot{R}(0)$ and $\dot{Q}(0)$ are non-zero, this initial condition is not a fixed point of DFA. To analyse initial alignment, we consider the first order term of $\dot{W}_2$. Using Eq. (28) with the derivatives at $t = 0$ (29), we obtain to linear order in t:

$$\dot{W}_2^k(t) = \frac{2}{\pi^2}\eta^2||\tilde{W}_2||^2 F_1^k t. \tag{30}$$

Crucially, this update is in the direction of the feedback vector F_1. DFA training thus constrains the student to initially grow in the direction of the feedback vector and align with it. This implies gradient alignment between BP and DFA and dictates into which of the many degenerate solutions in the energy landscape the student converges.

Plateau phase After the initial phase of learning with DFA where the test error decreases exponentially, similarly to BP, the student falls into a symmetric fixed point of the Eqs. (27) where the weights of a single student node are correlated to the weights of all the teacher nodes ((Saad & Solla, 1995a; Biehl & Schwarze, 1995; Engel & Van den Broeck, 2001)). The test error stays constant while the student is trapped in this fixed point. We can obtain an analytic expression for the order parameters under the assumption that the teacher first-layer weights are orthogonal ($T^{nm} = \delta_{nm}$). We set the teacher's second-layer weights to unity for notational simplicity ($\tilde{W}_2^m = 1$) and restrict to linear order in the learning rate η, since this is the dominant contribution to the learning dynamics at early times and on the plateau (Saad & Solla, 1995b). In the case where all components of the feedback vector are positive, the order parameters are of the form $Q^{kl} = q, R^{km} = r, W_2^k = w_2$ with:

$$q = \frac{1}{2K-1}, \quad r = \sqrt{\frac{q}{2}}, \quad w_2 = \sqrt{\frac{1+2q}{q(4+3q)}}. \tag{31}$$

If the components of the feedback vector are not all positive, we instead obtain $R^{km} = \mathrm{sgn}(F^k)r$, $W_2^k = \mathrm{sgn}(F^k)w_2$ and $Q^{kl} = \mathrm{sgn}(F^k)\,\mathrm{sgn}(F^l)q$. This shows that on the plateau the student is already in the configuration that maximises its alignment with F_1. Note that in all cases, the value of the test error reached at the plateau is the same for DFA and BP.

Memorisation phase and Asymptotic Fixed Point At the end of the plateau phase, the student converges to its final solution, which is often referred to as the *specialised* phase (Saad & Solla, 1995a; Biehl & Schwarze, 1995; Engel & Van den Broeck, 2001). The configuration of the order parameters is such that the student reproduces her teacher up to sign changes that guarantee the alignment between W_2 and F_1 is maximal, i.e. $\mathrm{sgn}(W_2^k) = \mathrm{sgn}(F_1^k)$. The final value of the test error of a student trained with DFA is the same as that of a student trained with BP on the same teacher.

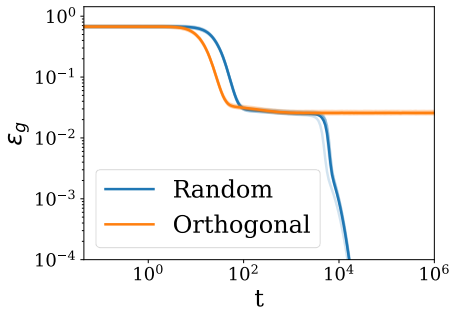


Figure 10. Test error of a sigmoidal student started with zero initial weights. The feedback vector F_1 is chosen random (blue) and orthogonal to the teacher's second layer weights $\tilde{W}_2$ (orange). *Parameters:* $\eta = 0.1, K = M = 2$.

Choice of the feedback vector In the main text, we saw how a wrong choice of feedback vector F_1 can prevent a ReLU student from learning a task. Here, we show that also for sigmoidal student, a *wrong* choice of feedback vector F_1 is possible. As Fig. 10 shows, in the case where the F_1 is taken orthogonal to the teacher second layer weights, a student whose weights are initialised to zero remains stuck on the plateau and is unable to learn. In contrast, when the F_1 is chosen with random i.i.d. components drawn from the standard normal distribution, perfect recovery is achieved.

C. Derivation of weight alignment

Since the network is linear, the update equations are (consider the first three layers only):

$$\delta W_1 = -\eta(F_1 e)x^T, \tag{32}$$

$$\delta W_2 = -\eta(F_2 e)(W_1 x)^\top, \tag{33}$$

$$\delta W_3 = -\eta(F_3 e)(W_2 W_1 x)^\top \tag{34}$$

First, it is straightforward to see that

$$W_1^t = -\eta \sum_{t'=0}^{t-1} F_1 e_{t'} x_{t'}^\top = F_1 A_1^t \tag{35}$$

$$A_1^t = -\eta \sum_{t'=0}^{t-1} e_{t'} x_{t'}^\top \tag{36}$$

This allows to calculate the dynamics of W_2^t:

$$\delta W_2^t = -\eta F_2 e_t (A_1^t x_t)^\top F_1^\top \tag{37}$$

$$W_2^t = -\eta \sum_{t'=0}^{t-1} F_2 e_t (A_1^{t'} x_{t'})^\top F_1^\top = F_2 A_2^t F_1^\top \tag{38}$$

$$A_2^t = -\eta \sum_{t'=0}^{t-1} e_{t'} (A_1^{t'} x_{t'})^\top = \eta^2 \sum_{t'=0}^{t-1} \sum_{t''=0}^{t'-1} (x_{t'} \cdot x_{t''}) e_{t'} e_{t''}^\top. \tag{39}$$

Which in turns allows to calculate the dynamics of W_3^t:

$$\delta W_3^t = -\eta F_3 e_t (F_2 A_2^{t'} F_1^\top F_1 A_1^{t'} x_t)^\top \tag{40}$$

$$W_3^t = -\eta \sum_{t'=0}^{t-1} F_3 e_{t'} (F_2 A_2^{t'} F_1^\top F_1 A_1^{t'} x_t)^\top = F_3 A_3^t F_2^\top \tag{41}$$

$$A_3^t = -\eta \sum_{t'=0}^{t-1} F_3 e_{t'} (A_2^{t'} F_1^\top F_1 A_1^{t'} x_{t'})^\top \tag{42}$$

$$= \eta^2 \sum_{t'=0}^{t-1} \sum_{t''=0}^{t'-1} (A_1^{t'} x_{t'}) \cdot (A_1^{t''} x_{t''}) e_{t'} e_{t''}^\top. \tag{43}$$

By induction it is easy to show the general expression:

$$A_1^t = -\eta \sum_{t'=0}^{t-1} e_{t'} x_{t'}^\top \tag{44}$$

$$A_2^t = \eta^2 \sum_{t'=0}^{t-1} \sum_{t''=0}^{t'-1} (x_{t'} \cdot x_{t''}) e_{t'} e_{t''}^\top \tag{45}$$

$$A_{l\geq 3}^t = \eta^2 \sum_{t,t'=0} (A_{l-2}^{t'} \dots A_1^{t'} x_{t'}) \cdot (A_{l-2}^{t''} \dots A_1^{t''} x_{t''}) e_{t'} e_{t''}^\top \tag{46}$$

Defining $A_0 \equiv \mathbb{I}_{n_0}$, one can rewrite this as in Eq. 15

$$A_{l\geq 2}^t = \eta^2 \sum_{t'=0}^{t-1} \sum_{t''=0}^{t'-1} (B_l^{t'} x_{t'}) \cdot (B_l^{t''} x_{t''}) e_{t'} e_{t''}^\top, \tag{47}$$

$$B_l = A_{l-2} \cdots A_0. \tag{48}$$

D. Impact of data structure

To study the impact of data structure on the alignment, the simplest setup to consider is that of Direct Random Target Projection (Frenkel et al., 2019). Indeed, in this case the error vector $e_t = -y_t$ does not depend on the prediction of the network: the dynamics become explicitly solvable in the linear case.

For concreteness, we consider the setup of (Lillicrap et al., 2016) where the targets are given by a linear teacher, $y = Tx$, and the inputs are i.i.d Gaussian. We denote the input and target correlation matrices as follows:

$$\mathbb{E}\left[xx^\top\right] \equiv \Sigma_x \in \mathbb{R}^{n_0 \times n_0}, \tag{49}$$

$$\mathbb{E}\left[TT^\top\right] \equiv \Sigma_y \in \mathbb{R}^{n_L \times n_L} \tag{50}$$

If the batch size is large enough, one can write $x_t x_t^\top = \mathbb{E}\left[xx^\top\right] = \Sigma_x$. Hence the dynamics of Eq. 9 become:

$$\delta W_1^t = -\eta (F_1 e_t) x_t^T = \eta F_1 T x_t x_t^\top = \eta F_1 T \Sigma_x \tag{51}$$

$$\delta W_2^t = -\eta (F_2 e_t)(W_1 x_t)^\top = \eta F_2 T \Sigma_x W_1^\top \tag{52}$$

$$= \eta^2 F_2 \left(T \Sigma_x^2 T^\top\right) F_1^\top \tag{53}$$

$$\delta W_3^t = -\eta (F_3 e_t)(W_2 W_1 x_t)^\top = \eta F_3 T \Sigma_x W_1^\top W_2^\top \tag{54}$$

$$= \eta^3 F_3 \left(T \Sigma_x^2 T^\top\right) \left(T \Sigma_x^2 T^\top\right) F_2^\top \tag{55}$$

From which we easily deduce $A_1^t = \eta T \Sigma_x t$, and the expression of the alignment matrices at all times:

$$A_{l\geq 2}^t = \eta^l \left(T \Sigma_x^2 T^\top\right)^{l-1} t \tag{56}$$

As we saw, GA depends on how well-conditioned the alignement matrices are, i.e. how different it is from the identity. To examine deviation from identity, we write $\Sigma_x = \mathbb{I}_{n_0} + \tilde{\Sigma}_x$ and $\Sigma_y = \mathbb{I}_{n_L} + \tilde{\Sigma}_y$, where the tilde matrices are small perturbations. Then to first order,

$$A_{l\geq 2}^t - I_{n_L} \propto (l-1) \left(\tilde{\Sigma}_y + 2T\tilde{\Sigma}_x T^\top\right) \tag{57}$$

Here we see that GA depends on how well-conditioned the input and target correlation matrices Σ_x and Σ_y are. In other words, if the different components of the inputs or the targets are correlated or of different variances, we expect GA to be hampered, observed in Sec. 4. Note that due to the $l-1$ exponent, we expect poor conditioning to have an even more drastic effect in deeper layers.

Notice that in this DRTP setup, the norm of the weights grows linearly with time, which makes DRTP inapplicable to regression tasks, and over-confident in classification tasks. It is clear in this case the the first layer learns the teacher, and the subsequent layers try to passively transmit the signal.

E. Details about the experiments

E.1. Direct Feedback Alignment implementation

We build on the Pytorch implementation of DFA implemented in (Launay et al., 2020), accessi-

ble at https://github.com/lightonai/dfa-scales-to-modern-deep-learning/tree/master/TinyDFA. Note that we do not use the shared feedback matrix trick introduced in this work. We sample the elements of the feedback matrix F_l from a centered uniform distribution of scale $1/\sqrt{n_l + 1}$.

E.2. Experiments on realistic datasets

We trained 4-layer MLPs with 100 nodes per layer for 1000 epochs using vanilla SGD, with a batch size of 32 and a learning rate of 10^{-4}. The datasets considered are MNIST and CIFAR10, and the activation functions are Tanh and ReLU.

We initialise the networks using the standard Pytorch initialization scheme. We do not use any momentum, weight decay, dropout, batchnorm or any other bells and whistles. We downscale all images to 14×14 pixels to speed up the experiments. Results are averaged over 10 runs.

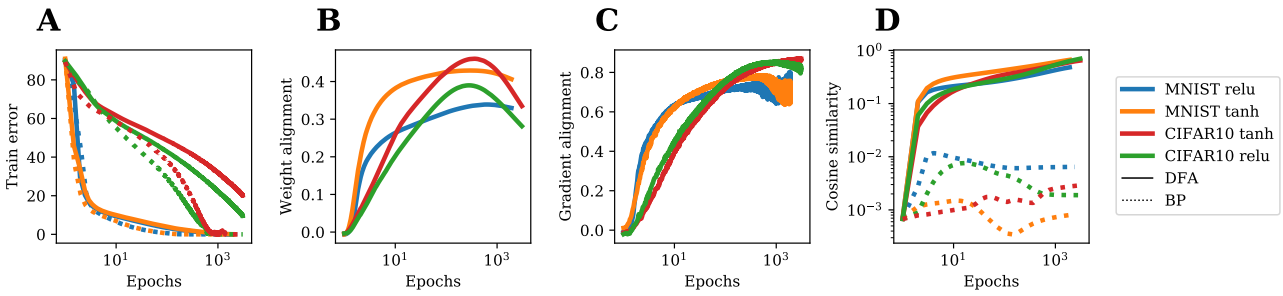
For completeness, we show in Fig. 11 the results in the main text for 4 different levels of label corruption. The transition from Alignment phase to Memorisation phase can clearly be seen in all cases from the drop in weight alignment. Three important remarks can be made:

- **Alignment phase**: Increasing label corruption slows down the early increase of weight alignment, as noted in Sec. 4.1.
- **Memorization phase**: Increasing label corruption makes the datasets harder to fit. As a consequence, the network needs to give up more weight alignment in the memorization phase, as can be seen from the sharper drop in the weight alignment curves.
- **Transition point**: the transition time between the Alignement and Memorization phases coincides with the time at which the training error starts to decrease sharply (particularly at high label corruption), and is hardly affected by the level of label corruption.

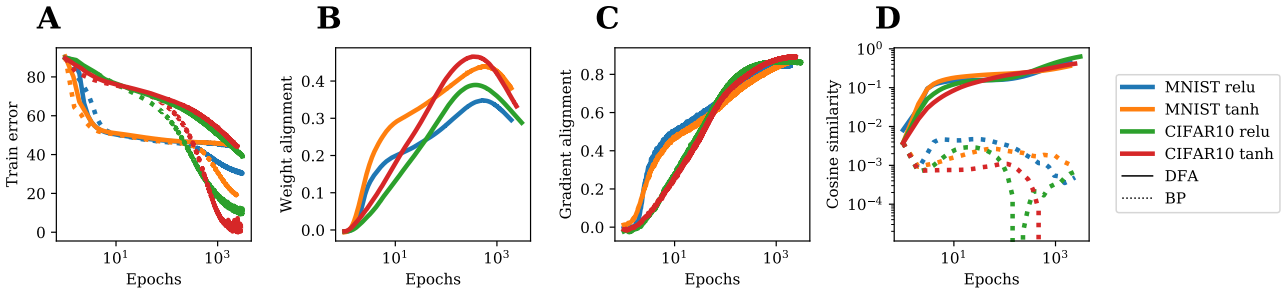
E.3. Experiment on the structure of targets

We trained a 3-layer linear MLP of width 100 for 1000 epochs on the synthetic dataset described in the main text, containing 10^4 examples. We used the same hyperparameters as for the experiment on nonlinear networks. We choose 5 values for α and β: 0.2, 0.4, 0.6, 0.8 and 1.

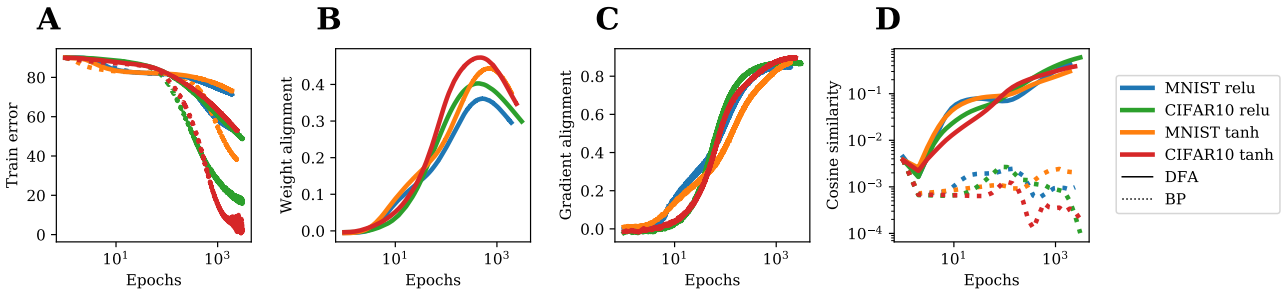
In Fig. 12, we show the dynamics of weight alignment for both ReLU and Tanh activations. We again see the Align-then-Memorise process distinctly. Notice that decreasing α and β hampers both the mamixmal weight alignment (at the end of the alignment phase) and the final weight alignment (at the end of the memorisation phase).



(a) No label corruption

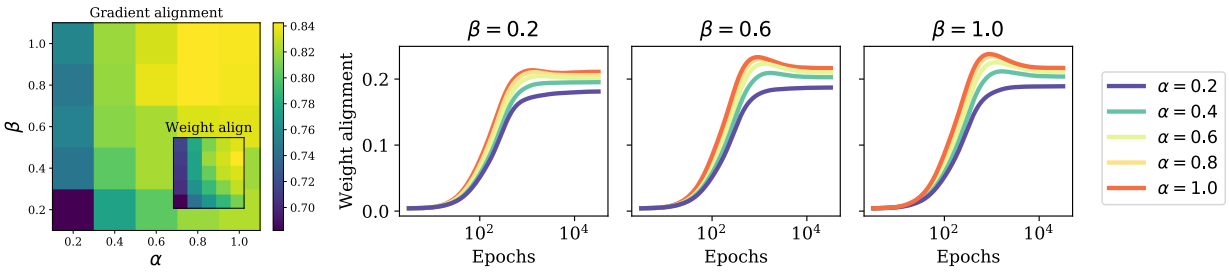


(b) 50% label corruption

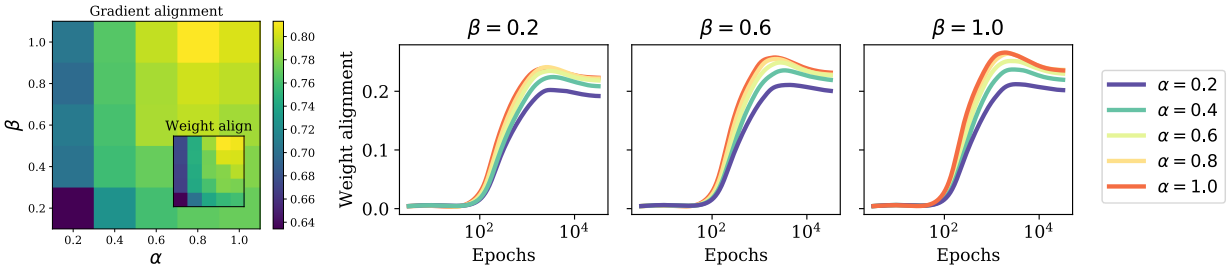


(c) 90% label corruption

Figure 11. Effect of label corruption on training observables. **A**: Training error. **B** and **C**: Weight and gradient alignment, as defined in the main text. **D**: Cosine similarity of the weight during training.



(a) ReLU



(b) Tanh

Figure 12. WA is hampered when the output dimensions are correlated ($\beta < 1$) or of different variances ($\alpha < 1$).