Supplemental Material for "Statistical mechanics of transfer learning in fully-connected networks"

I. SETTING AND NOTATION

The output of a one-hidden layer fully connected network given a data point $x \in R^{N_0}$ is:

$$\phi(x) = \frac{1}{\sqrt{N_1}} \sum_{k=1}^{N_1} v_k \sigma\left(\sum_{i=1}^{N_0} w_{ki} x_i\right) \tag{1}$$

with σ a non-linear function. We will use erf as an example of a symmetric, saturating function: generalization to other nonlinearities is straightforward. Given a dataset of inputs and outputs $\mathcal{D} = \{X, y\}$, with X the usual design matrix $X = (X)_{\mu i}$, the loss function reads:

$$\mathcal{L}(\theta, \mathcal{D}) = \frac{1}{2} \sum_{x,y \in \mathcal{D}} \left[y - \phi(\theta, x)\right]^2 \tag{2}$$

where $\theta \equiv \{w, v\}$ is a shorthand for the collection of first and second layer weights.

We consider the learning problem of a one-hidden layer *target* network whose first-layer weights are coupled via a parameter γ to those of a previously trained *source* network. Let us consider three datasets X_s, X_t, X_τ of size P_s, P_t, P_τ, respectively the training set for the source and target task, and test set for the target task, with their respective outputs y_s, y_t, y_τ.

Performing the quenched average over the source posterior weights of the log-partition function of the transfer weight, we get to the following expression:

$$N_1 f = \frac{1}{Z_s} \int d\mu_s(w_s)\, e^{-\beta_s \mathcal{L}_s(w_s)} \log \int d\mu_t(w_t)\, e^{-\beta_t \mathcal{L}_t(w_t) - \frac{\gamma}{2}\|w_s - w_t\|^2} \tag{3}$$

where

$$d\mu_{s/t}(w) = dw e^{-\frac{\lambda_{s/t,1}}{2}\|w\|^2 - \frac{\lambda_{s/t,2}}{2}\|v\|^2} \tag{4}$$

and

$$Z_s(\beta_s) = \int d\mu_s(w_s)\, e^{-\beta_s \mathcal{L}_s(w_s)} \tag{5}$$

is the source partition function. To deal with the log, we make use of the replica trick:

$$\log Z = \lim_{n \to 0} \partial_n Z^n \tag{6}$$

and we write the transfer free-entropy in terms of replicated variables as

$$N_1 f = \frac{1}{Z_s(\beta_s)} \lim_{n \to 0} \partial_n \int d\mu_s(w_s) \prod_{a=1}^{n} d\mu_t(w_t^a)\, e^{-\beta_s \mathcal{L}_s(w_s)} \\ e^{-\beta_t \sum_{a=1}^n \mathcal{L}_t(\{w_t^a\}) - \frac{\gamma}{2}\sum_a \|w_s - w_t^a\|^2}. \tag{7}$$

Eq. (7) represents a single-instance generalization of the classic disordered-averaged Franz-Parisi approach, originally developed to study metastable states in spin-glasses. We can employ f to compute the posterior average of relevant quantities, such as training error, weight norms and distance across weight matrices, using simple differentiation:

$$\epsilon_t \equiv \langle \mathcal{L}_t \rangle = -\frac{N_1}{P_t} \partial_{\beta_t} f\,, \tag{8}$$

$$\|w_t\|^2 = -2N_1 \partial_\lambda f\,, \tag{9}$$

$$\|w_s - w_t\|^2 = 2N_1 \partial_\gamma f\,. \tag{10}$$

The calculation of the generalization error is slightly more involved: we introduce and additional term $\beta_\tau \mathcal{L}_\tau(\{w_t\})$ in the target Hamiltonian, where $\mathcal{L}_\tau(\{w_t\}) = \frac{1}{2}\sum_{\mu=1}^{P_\tau} \left[y_\tau^\mu - \phi(\theta_t, x_\tau^\mu)\right]^2$ is the loss computed on a test set composed of P_τ patterns, and evaluate the test error with the expression $\epsilon_\tau \equiv \langle \mathcal{L}_\tau \rangle = -\frac{N_1}{P_\tau} \partial_{\beta_\tau} f|_{\beta_\tau = 0}$.

Generalization to a one-hidden layer convolutional neural network The output of a one-hidden layer convolutional neural network (CNN) with filters of size M and stride S can be written as:

$$\phi^{CNN}\left(x;\{w,v\}\right)=\frac{1}{\sqrt{N_c}}\sum_{i=1}^{N_0/S}\sum_{c}^{N_c}v_i^c\sigma\left(\frac{1}{\sqrt{M}}\sum_m w_{cm}x_{Si+m}\right). \tag{11}$$

Very few changes are in order to generalize our calculation to the case of a shallow CNN, which we will thus highlight along the way.

A. Notation

We use the 0 index for all weights and parameters involving the source, thus denoting the prior inverse variances in each layer l as $\lambda_l^0=\lambda_{s,l}$ and $\lambda_l^a\equiv\lambda_{t,l}$ for $a>0$. We thus treat (a,μ) as a multi-index over a construction with concatenated source and replicated target inputs $[X_s,\underbrace{X_t,...X_t}_{n}]$, and trust the context to make the summation over μ clear. The same is done for the inverse temperatures, i.e. we have $\beta_\mu^0=\beta_s$ and $\beta_\mu^a=\beta_t$ for $a>0$. All sums over a will implicitly run from 0 to n, unless specified by the subscript.

We will denote by $\mathbb{1}_m$ the identity matrix in dimension m, the vector $\left(e_m^j\right)_i=\delta_{ij}$ is the canonical base vector and $\left(\mathbb{l}_m\right)_i=1$ the constant vector of value 1. We will usually drop the subscript when the dimension m is implied by the context. We denote by $\mathcal{N}\left(h;m,C\right)$ a Gaussian distribution with mean m and covariance C over the vector h. All subleading factors will be discarded to reduce clutter.

II. TRANSFER LEARNING IN A ONE-HIDDEN LAYER NETWORKS

Our aim is to compute the following replicated partition function:

$$\begin{aligned}Z^n=&\int\prod_{ki}dw_{ki}e^{-\frac{1}{2}\sum_a\lambda_1^a\|w^a\|^2-\frac{\gamma}{2}\sum_a\left\|w^0-w^a\right\|^2}\\&\int\prod_{ak}dv_k^a e^{-\frac{1}{2}\sum\lambda_2^a\|v^a\|^2-\frac{1}{2}\sum_{a\mu}\beta_\mu^a\left(\sum_k\frac{v_k^a}{\sqrt{N_1}}\sigma\left(\frac{1}{\sqrt{N_0}}\sum_i w_{ki}^a x_{\mu i}^a\right)-y_\mu^a\right)^2}.\end{aligned} \tag{12}$$

We expect that the final form will read

$$Z^n\sim\int\mathcal{D}\mathcal{Q}\mathcal{D}\bar{\mathcal{Q}}e^{\frac{N_1}{2}nS(\mathcal{Q},\bar{\mathcal{Q}})}, \tag{13}$$

with $\mathcal{Q}$ and $\bar{\mathcal{Q}}$ a set of order parameters whose values will be determined by saddle-point equations. The $\mathcal{O}\left(1\right)$ component $e^{\frac{N_1}{2}S_s(Q_s,\bar{Q}_s)}$ of the replicated partition function Z^n only depends on source order parameters and will cancel out with the term Z_s^{-1} at the saddle point.

A. Integrating first layer weights

Introducing the definition for the first-layer replicated pre-activations $h_{\mu k}^a=\frac{1}{\sqrt{N_0}}\sum_i w_{ki}^a x_i^{\mu a}$, we have:

$$\begin{aligned}Z^n=&\int\prod_{ki}w_{ki}e^{-\frac{1}{2}\sum_a\lambda_1^a\|w\|^2-\frac{\gamma}{2}\sum_a\left\|w^0-w^a\right\|^2}\\&\int\prod_{\mu ak}d\bar{h}_{\mu k}^a dh_{\mu k}^a e^{i\sum_{a\mu k}\bar{h}_{\mu k}^a\left(\bar{h}_k^a-\frac{1}{\sqrt{N_0}}\sum_i w_{ki}^a x_{\mu i}^a\right)}\\&\int\prod_{ak}dv_k^a e^{-\frac{\lambda_2^a}{2}\sum_a\|v^a\|^2-\frac{1}{2}\sum_{a\mu}\beta_\mu^a\left(\sum_k\frac{v_k^a}{\sqrt{N_1}}\sigma\left(h_{\mu k}^a\right)-y_\mu^a\right)^2}.\end{aligned} \tag{14}$$

Let us isolate the dependence over the first layer pre-activations h in the function ψ:

$$\psi(h) = \int \prod_{a\mu k} d\bar{h}^a_{\mu k} e^{i \sum_{a\mu k} \bar{h}^a_{\mu k} h^a_{\mu k}} \int \mathcal{D}w \prod_{ki} e^{-\frac{1}{2}\sum_{ab}\Lambda_1^{ab} w^a_{ki} w^b_{ki} - \frac{i}{\sqrt{N_0}}\sum_{a\mu} w^a_{ki}\bar{h}^a_{\mu k} x^a_{\mu i}} \tag{15}$$

where we have defined the following coupling matrix:

$$\Lambda_1 = \begin{pmatrix} \lambda_{s,1} & -\gamma & -\gamma & ... & -\gamma \\ -\gamma & \lambda_{t,1} & 0 & ... & 0 \\ -\gamma & 0 & \lambda_{t,1} & ... & ... \\ ... & ... & ... & ... & 0 \\ -\gamma & 0 & ... & 0 & \lambda_{t,1} \end{pmatrix} \tag{16}$$

and $\mathcal{D}w \equiv \prod_{aki} dw^a_{ki}$. We thus write compactly:

$$Z^n = \int \mathcal{D}h \psi(h) \int \prod_{ak} dv^a_l e^{-\frac{1}{2}\sum_{a\mu}\beta^a_\mu \left(\sum_k \frac{v^a_k}{\sqrt{N_1}}\sigma\left(h^a_{\mu k}\right) - y^a_\mu\right)^2 - \frac{\lambda_2^a}{2}\sum_a \|v^a\|^2} \tag{17}$$

and integrate over the weights

$$\tag{18}$$

with the definitions

$$\bar{q}^a_{ki}\left(\bar{h}\right) = \frac{1}{\sqrt{N_0}}\sum_\mu \bar{h}^a_{\mu k} x^a_{\mu i} \tag{19}$$

$$\Delta_1 = e^{-\frac{N_1 N_0}{2}\log\det\Lambda}\,. \tag{20}$$

We can write Z^n in terms of the zero-mean, Gaussian distributed variables $h^a_{\mu k}$, whose covariance matrices $\tilde{C}$ read, for each k:

$$\tilde{C}^{ab}_{\mu\nu} = \left\langle h^a_{\mu k} h^b_{\nu k}\right\rangle = \Lambda^{-1}_{ab} C^{ab}_{\mu\nu} \tag{21}$$

where $C^{ab}_{\mu\nu}$ are replicated input covariances:

$$C^{ab}_{\mu\nu} = \frac{1}{N_0}\sum_{i=1}^{N_0} x^a_{\mu i} x^b_{\mu i}\,. \tag{22}$$

We thus have:

$$Z^n = \Delta_1 \int \prod_k \mathcal{D}h_k \mathcal{D}v_k \mathcal{N}\left(h_k; 0, \tilde{C}\right) e^{-\frac{\lambda_2^a}{2}\sum_a \|v^a\|^2 - \frac{1}{2}\sum_{a\mu}\beta^a_\mu\left(\sum_k \frac{v^a_k}{\sqrt{N_1}}\sigma\left(h^a_{\mu k}\right) - y^a_\mu\right)^2} \tag{23}$$

with $\mathcal{D}h_k\mathcal{D}v_k \equiv \prod_{a\mu} dh^a_{\mu k} \prod_a dv^a_k$.

Generalization to a 1-hl convolutional neural network In the case of a convolutional layer, we would operate in the same manner and find $\bar{q}^a_{cm} = \frac{1}{\sqrt{M}}\sum_{\mu i}\bar{h}^a_{\mu c i} x^\mu_{Si+m}$. Note that variables $\bar{q}$ carry an additional index c for each patch, so that $\bar{\mathcal{Q}}_{ab}$ is an $N_c \times N_c$ dimensional matrices, with N_c the number of patches. The inter-patch input covariances reads:

$$\tilde{C}^{ab,ij}_{\mu\nu} = \left\langle h^a_{\mu i} h^b_{\nu j}\right\rangle = \left(\Lambda^{-1}\right)^{ab} C^{ab,ij}_{\mu\nu} \tag{24}$$

$$C^{ab,ij}_{\mu\nu} = \frac{1}{M}\sum_m x^{a,\mu}_{Si+m} x^{b,\nu}_{Sj+m}\,. \tag{25}$$

B. Integrating readout weights

Introducing the definition of the readout outputs $s_\mu^a = \sum_k \frac{v_k^a}{\sqrt{N_1}} \sigma\left(h_{\mu k}^a\right)$ with appropriate δ functions, we obtain an expression of the form

$$Z^n = \Delta_1 \int \mathcal{D}s \psi\left(s\right) \tag{26}$$

where

$$\begin{aligned}\psi\left(s\right) = \int \mathcal{D}s\mathcal{D}\bar{s} \prod_k \mathcal{D}h_k \mathcal{D}v_k \mathcal{N}\left(h_k; 0, \tilde{C}\right) \\ \prod_k e^{-\frac{1}{2}\sum_a \lambda_2^a (v_k^a)^2} e^{-\frac{i}{\sqrt{N_1}} \sum_{a\mu} \bar{s}_\mu^a \sum_k v_k^a \sigma\left(h_{\mu k}^a\right)} \\ e^{i\sum_{a\mu} \bar{s}_\mu^a s_\mu^a - \frac{1}{2}\sum_{a\mu} \beta_\mu^a \left(s_\mu^a - y_\mu^a\right)^2}\end{aligned} \tag{27}$$

with the shorthand $\mathcal{D}s\mathcal{D}\bar{s} = \prod_{a\mu k} ds_{\mu k}^a d\bar{s}_{\mu k}^a$. After a straghtforward integration over the uncoupled second-layer weights, we have:

$$\begin{aligned}\psi\left(s\right) = \int \mathcal{D}s\mathcal{D}\bar{s} \prod_k \mathcal{D}h_k \mathcal{D}v_k \mathcal{N}\left(h_k; 0, \tilde{C}\right) \\ \Delta_2 \prod_k e^{-\frac{i}{N_1}\sum_{a\mu} \frac{\bar{s}_\mu^a \bar{s}_\nu^a}{\lambda_2^a} \sigma\left(h_{\mu k}^a\right)\sigma\left(h_{\mu k}^a\right)} \\ e^{i\sum_{a\mu} \bar{s}_\mu^a s_\mu^a - \frac{1}{2}\sum_{a\mu} \beta_\mu^a \left(s_\mu^a - y_\mu^a\right)^2}\end{aligned} \tag{28}$$

where we have introduced $\Delta_2 = e^{-\frac{N_1}{2}\log \Lambda_2}$ using the second-layer coupling matrix

$$\Lambda_2 = \begin{pmatrix} \lambda_{s,2} & 0 & 0 & ... & 0 \\ 0 & \lambda_{t,2} & 0 & ... & 0 \\ 0 & 0 & \lambda_{t,2} & ... & ... \\ ... & ... & ... & ... & 0 \\ 0 & 0 & ... & 0 & \lambda_{t,2} \end{pmatrix} . \tag{29}$$

We are now ready to employ the factorization over the first hidden layer index k and consider the Gaussian variables h_μ^a. Following the same strategy of Refs. [1, 2], we perform a self-consistent Gaussian approximation on the set of variables $\bar{q}^a = \frac{1}{\sqrt{\lambda_2^a N_1}} \sum_\mu \bar{s}_\mu^a \sigma\left(h_\mu^a\right)$ in replica space, with order-parameter covariance matrix:

$$\bar{\mathcal{Q}}^{ab} = \left\langle \bar{q}^a \bar{q}^b \right\rangle = \frac{1}{N_1 \sqrt{\lambda_2^a \lambda_2^b}} \sum_{\mu\nu} \bar{s}_\mu^a K_{\mu\nu}^{ab} \bar{s}_\nu^b \tag{30}$$

and kernels:

$$K_{\mu\nu}^{ab} = \left\langle \sigma\left(h_\mu^a\right) \sigma\left(h_\nu^a\right) \right\rangle_{\mathcal{N}\left(h;0,\tilde{C}\right)} . \tag{31}$$

The relevant replicated kernels are:

$$\tilde{K}_{s,\mu\nu} = \left\langle \sigma\left(h_\mu^0\right) \sigma\left(h_\nu^0\right) \right\rangle_{\mathcal{N}\left(h;0,\tilde{C}\right)} \tag{32}$$

$$K_{st,\mu\nu} = \left\langle \sigma\left(h_\mu^0\right) \sigma\left(h_\nu^a\right) \right\rangle_{\mathcal{N}\left(h;0,\tilde{C}\right)} \qquad a > 0 \tag{33}$$

$$K_{t,\mu\nu} = \left\langle \sigma\left(h_\mu^a\right) \sigma\left(h_\nu^a\right) \right\rangle_{\mathcal{N}\left(h;0,\tilde{C}\right)} \qquad a > 0 \tag{34}$$

$$K_{tt,\mu\nu} = \left\langle \sigma\left(h_\mu^a\right) \sigma\left(h_\nu^b\right) \right\rangle_{\mathcal{N}\left(h;0,\tilde{C}\right)} \qquad a \neq b; a, b > 0 \tag{35}$$

with order parameters:

$$\bar{Q}_s = \bar{\mathcal{Q}}^{00} \tag{36}$$
$$\bar{Q}_{st} = \bar{\mathcal{Q}}^{0a} \quad a > 0 \tag{37}$$
$$\bar{Q}_t = \bar{\mathcal{Q}}^{aa} \quad a > 0 \tag{38}$$
$$\bar{Q}_{tt} = \bar{\mathcal{Q}}^{ab} \quad a \neq b;\ a, b > 0\,. \tag{39}$$

Note that, at this stage, the kernels explicitly depend on the replica number n. Introducing the definitions of the order parameters with the help of appropriate δ functions and conjugate parameters $\mathcal{Q}$, we finally obtain:

$$Z^n = \Delta_1 \Delta_2 \int \mathcal{D}\mathcal{Q}\mathcal{D}\bar{\mathcal{Q}} e^{\frac{N_1}{2}\operatorname{Tr}\left(\tilde{\Lambda}_2 \mathcal{Q}\bar{\mathcal{Q}}\right) - \frac{1}{2}\log\det\left(\mathbb{1}+\bar{\mathcal{Q}}\right)} \int \mathcal{D}s\mathcal{D}\bar{s} e^{i\bar{s}^T s - \frac{1}{2}\bar{s}^T \mathcal{K}\bar{s} - \frac{1}{2}(s-y)^T B(s-y)} \tag{40}$$

where $\mathcal{K}$ is the renormalized kernel $\mathcal{K}^{ab}_{\mu\nu} = \mathcal{Q}^{ab} K^{ab}_{\mu\nu}$ and $\left(\tilde{\Lambda}_2\right)_{ab} = \sqrt{\lambda_2^a \lambda_2^b}$. To simplify notation, we introduced a replicated vector $(s)_{a\mu} = s^a_\mu$ for the readout variables and targets y^a_μ, and correspondingly for the diagonal matrix $B^{ab}_{\mu\nu}$ containing the inverse temperatures. More concretely, we have:

$$B = \begin{pmatrix} \beta_s \mathbb{1}_{P_s} & 0 & 0 & ... & 0 \\ 0 & \beta_t \mathbb{1}_{P_t} & 0 & ... & 0 \\ 0 & 0 & \beta_t \mathbb{1}_{P_t} & & ... \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & & \beta_t \mathbb{1}_{P_t} \end{pmatrix} \tag{41}$$

Integrating over $\bar{s}$ is straightforward:

$$Z^n = \Delta_1 \Delta_2 e^{-\frac{1}{2} y^T B y} \int \mathcal{D}\mathcal{Q}\mathcal{D}\bar{\mathcal{Q}} e^{\frac{N_1}{2}\operatorname{Tr}\left(\tilde{\Lambda}_2 \mathcal{Q}\bar{\mathcal{Q}}\right) - \frac{1}{2}\log\det\left(\mathbb{1}+\bar{\mathcal{Q}}\right)} e^{-\frac{1}{2} s^T \left(B + \mathcal{K}^{-1}\right) s + s^T B y - \frac{1}{2}\log\det \mathcal{K}}\,. \tag{42}$$

Further integrating over s, we find the final form valid for integer number replicas

$$Z^n = \int \mathcal{D}\mathcal{Q}\mathcal{D}\bar{\mathcal{Q}} e^{\frac{N_1}{2} S(\mathcal{Q}, \bar{\mathcal{Q}})} \tag{43}$$

where the finite-n action is given by:

$$S = -N_0 \log\det\Lambda_1 - \log\det\Lambda_2 + \operatorname{Tr}\left(\tilde{\Lambda}_2 \mathcal{Q}\bar{\mathcal{Q}}\right) - \log\det\left(\mathbb{1} + \bar{\mathcal{Q}}\right) - \frac{1}{N_1}\log\det B\Sigma - \frac{1}{N_1} y^T \Sigma^{-1} y, \tag{44}$$

with

$$\Sigma = B^{-1} + \mathcal{K} \tag{45}$$

Generalization to a one-hidden layer convolutional neural network In the case of a convolutional network, we would operate in the same manner and find $\bar{q}^{CNN,a}_{0,cm} = \frac{1}{\sqrt{M}} \sum_{\mu i} \bar{h}^a_{\mu ci} x^\mu_{Si+m}$, from which the replicated renormalized local kernel $\mathcal{K}^{ab}_{\mu\nu} = \sum_{ij} \mathcal{Q}^{ab}_{ij} K^{ab}_{ij,\mu\nu}$ is found.

C. Source and target action

As explained at the beginning of this section, the source-only action S_s can be easily obtained from the $\mathcal{O}(1)$ terms in n:

$$S_s = \lambda_{s,2} Q_s \bar{Q}_s - \log\left(1+\bar{Q}_s\right) - \frac{1}{N_1}\log\det\beta_s\Sigma_s - \frac{1}{N_1} {y_s}^T \Sigma_s^{-1} y_s + \\ - N_0 \log\det\lambda_{s,1} - \log\det\lambda_{s,2} \tag{46}$$

with

$$\Sigma_s = \frac{\mathbb{1}}{\beta_s} + Q_s K_s \tag{47}$$

and K_s the source-only first-layer kernel:

$$K_{s,\mu\nu} = \left\langle \sigma\left(h_\mu\right)\sigma\left(h_\nu\right)\right\rangle_{\mathcal{N}\left(h;0,\frac{C_s}{\lambda_{s,1}}\right)} \tag{48}$$

The values of Q_s, $\bar{Q}_s$ are taken from the source-only saddle-point equations involved in determining Z_s, as in the classic Franz-Parisi approach.

The genuine transfer action is obtained collecting the $\mathcal{O}(n)$ terms. The algebraic details involved in the diagonalization of the $\mathcal{Q}$ and Σ matrices are collected in section II F 0 a. Introducing the modified kernel matrices

$$\Sigma_t = \frac{\mathbb{1}}{\beta_t} + Q_t K_t \tag{49}$$

$$\Delta\Sigma_t = \Sigma_t - Q_{tt} K_{tt} \tag{50}$$

and taking the $n \to 0$ limit, we finally have:

$$S = \Psi_T - \Psi_\Delta - \frac{1}{N_1}\Psi_S - \frac{1}{N_1}\Psi_{\mathcal{L}} \tag{51}$$

where

$$\Psi_T = 2\sqrt{\lambda_{s,2}\lambda_{t,2}}\bar{Q}_{st}Q_{st} + \lambda_{t,2}\left(\bar{Q}_t Q_t - \bar{Q}_{tt}Q_{tt}\right) \\ - \log\left(1+\bar{Q}_t-\bar{Q}_{tt}\right) - \frac{\left(1+\bar{Q}_s\right)\bar{Q}_{tt} - \bar{Q}_{st}^2}{\left(1+\bar{Q}_s\right)\left(1+\bar{Q}_t-\bar{Q}_{tt}\right)} \tag{52}$$

$$\Psi_\Delta = N_0\left(\log\tilde{\lambda} + \frac{\lambda_{t,1}\gamma}{\tilde{\lambda}\lambda_{s,1}}\right) + \log\det\lambda_{t,2} \tag{53}$$

$$\Psi_S = \log\det\beta_t\Delta\Sigma_t + Q_{tt}\,\mathrm{Tr}\left(\Delta\Sigma_t^{-1}K_{tt}\right) - Q_{st}^2\,\mathrm{Tr}\left(\Sigma_s^{-1}K_{st}\Delta\Sigma_t^{-1}K_{st}^T\right) \tag{54}$$

$$\Psi_{\mathcal{L}} = y_t^T\Delta\Sigma_t^{-1}y_t^T - 2Q_{st}y_s^T\Sigma_s^{-1}K_{st}\Delta\Sigma_t^{-1}y_t + Q_{st}^2 y_s^T\Sigma_s^{-1}K_{st}\Delta\Sigma_t^{-1}K_{st}^T\Sigma_s^{-1}y_s \tag{55}$$

with $\tilde{\lambda} = \lambda_{t,1} + \gamma$. The three relevant kernels for the target action are the following:

$$K_{st,\mu\nu} = \left\langle \sigma\left(h_\mu\right)\sigma\left(h_\nu\right)\right\rangle_{\mathcal{N}\left(h;0,\tilde{C}_{st}\right)} \tag{56}$$

$$K_{t,\mu\nu} = \left\langle \sigma\left(h_\mu\right)\sigma\left(h_\nu\right)\right\rangle_{\mathcal{N}\left(h;0,\tilde{C}_{t}\right)} \tag{57}$$

$$K_{tt,\mu\nu} = \left\langle \sigma\left(h_\mu\right)\sigma\left(h_\nu\right)\right\rangle_{\mathcal{N}\left(h;0,\tilde{C}_{tt}\right)} \tag{58}$$

where the source-target and target modified covariance matrices are given by

$$\tilde{C}_{st} = \frac{\gamma}{\tilde{\lambda}\lambda_{s,1}}C_{st} \tag{59}$$

$$\tilde{C}_t = \frac{1}{\tilde{\lambda}}\left(1 + \frac{\gamma^2}{\tilde{\lambda}\lambda_{s,1}}\right)C_t \tag{60}$$

and the inter-replica target kernel is computed using a matrix $\tilde{C}_{tt}$ with entries

$$\tilde{C}_{tt,\mu\nu} = \delta_{\mu\nu}\tilde{C}_{t,\mu\mu} + \left(1-\delta_{\mu\nu}\right)\frac{\gamma^2}{\tilde{\lambda}^2\lambda_{s,1}}C_{t,\mu\nu} \tag{61}$$

where $\tilde{\lambda} \equiv \lambda_{t,1} + \gamma$. In the case of erf activation, we use the well known expression for the NNGP kernel [3]

$$\left\langle \sigma\left(h_\mu\right)\sigma\left(h_\nu\right)\right\rangle_{\mathcal{N}(h;0,C)} = \frac{2}{\pi}\arcsin\left(\frac{2C_{\mu\nu}}{\sqrt{\left(1+2C_{\mu\mu}\right)\left(1+2C_{\nu\nu}\right)}}\right). \tag{62}$$

D. SP equations

The SP equations for the order parameters read

$$\sqrt{\lambda_{s,2}\lambda_{t,2}}Q_{st} + \frac{\bar{Q}_{st}}{\left(1+\bar{Q}_s\right)\left(1+\bar{Q}_t-\bar{Q}_{tt}\right)} = 0 \tag{63}$$

$$\lambda_{t,2}Q_t = \frac{\bar{Q}_{st}^2 + \left(1+\bar{Q}_s\right)\left(1+\bar{Q}_t-2\bar{Q}_{tt}\right)}{\left(1+\bar{Q}_s\right)\left(1+\bar{Q}_t-\bar{Q}_{tt}\right)^2} \tag{64}$$

$$\lambda_{t,2}Q_{tt} = \frac{\bar{Q}_{st}^2 - \left(1+\bar{Q}_s\right)\bar{Q}_{tt}}{\left(1+\bar{Q}_s\right)\left(1+\bar{Q}_t-\bar{Q}_{tt}\right)^2} \tag{65}$$

whereas for the conjugate variables we have:

$$\begin{aligned} &N_1\sqrt{\lambda_{s,2}\lambda_{t,2}}\bar{Q}_{st} = -Q_{st}\,\mathrm{Tr}\left(\Sigma_s^{-1}K_{st}\Delta\Sigma_t^{-1}K_{st}^T\right) + Q_{st}y_s^T\Sigma_s^{-1}K_{st}\Delta\Sigma_t^{-1}K_{st}^T\Sigma_s^{-1}y_s \\ &- Q_{st}y_s^T\Sigma_s^{-1}K_{st}\Delta\Sigma_t^{-1}y_t \end{aligned} \tag{66}$$

$$\begin{aligned} &N_1\lambda_{t,2}\bar{Q}_t = \mathrm{Tr}\left(\Delta\Sigma_t^{-1}K_t\right) - Q_{tt}\,\mathrm{Tr}\left(\Delta\Sigma_t^{-1}K_t\Delta\Sigma_t^{-1}K_{tt}\right) \\ &+ Q_{st}^2\,\mathrm{Tr}\left(\Sigma_s^{-1}K_{st}\Delta\Sigma_t^{-1}K_t\Delta\Sigma_t^{-1}K_{st}^T\right) \\ &- Q_{st}^2y_s^T\Sigma_s^{-1}K_{st}\Delta\Sigma_t^{-1}K_t\Delta\Sigma_t^{-1}K_{st}^T\Sigma_s^{-1}y_s + 2Q_{st}y_s^T\Sigma_s^{-1}K_{st}\Delta\Sigma_t^{-1}K_t\Delta\Sigma_t^{-1}y_t \\ &- y_t^T\Delta\Sigma_t^{-1}K_t\Delta\Sigma_t^{-1}y_t^T \end{aligned} \tag{67}$$

$$\begin{aligned} &N_1\lambda_{t,2}\bar{Q}_{tt} = -Q_{tt}\,\mathrm{Tr}\left(\Delta\Sigma_t^{-1}K_{tt}\Delta\Sigma_t^{-1}K_{tt}\right) + Q_{st}^2\,\mathrm{Tr}\left(\Sigma_s^{-1}K_{st}\Delta\Sigma_t^{-1}K_{tt}\Delta\Sigma_t^{-1}K_{st}^T\right) + \\ &- Q_{st}^2y_s^T\Sigma_s^{-1}K_{st}\Delta\Sigma_t^{-1}K_{tt}\Delta\Sigma_t^{-1}K_{st}^T\Sigma_s^{-1}y_s + 2Q_{st}y_s^T\Sigma_s^{-1}K_{st}\Delta\Sigma_t^{-1}K_{tt}\Delta\Sigma_t^{-1}y_t \\ &- y_t^T\Delta\Sigma_t^{-1}K_{tt}\Delta\Sigma_t^{-1}y_t^T\,. \end{aligned} \tag{68}$$

E. Norm of the weights and generalization error

The norm of the second-layer weights can be easily obtained differentiating the action with respect to $\lambda_{t,2}$:

$$N_1\left\langle \|v\|^2 \right\rangle = \frac{1}{\lambda_{t,2}} + Q_{tt}\bar{Q}_{tt} - Q_t\bar{Q}_t - \sqrt{\frac{\lambda_{s,2}}{\lambda_{t,2}}}Q_{st}\bar{Q}_{st} \tag{69}$$

As explained in section I, we can compute the generalization error by differentiating the free-energy with respect to an additional fictitious temperature β_τ, coupled with a loss computed on a test set. This can be easily done by carrying out the previous calculation with an extended target dataset obtained by concatenating the input matrices $X_{\tilde{t}} = [X_t, X_\tau]$ and output vectors $y_{\tilde{t}} = [y_t, y_\tau]$. Accordingly, each target block in the matrix B is extended so as to contain $P_t + P_\tau$ diagonal entries from the vector $\beta_\mu = (\underbrace{\beta_t, ..., \beta_t}_{P_t}, \underbrace{\beta_\tau, ..., \beta_\tau}_{P_\tau})$. In what follows, the kernels $K_{s\tilde{t}}$, $K_{\tilde{t}}$ and $K_{\tilde{t}\tilde{t}}$ are obtained from equations (56–61) using covariance matrices from the extended dataset $X_{\tilde{t}}$. Calling $\Delta K = Q_t K_{\tilde{t}} - Q_{tt}K_{\tilde{t}\tilde{t}}$ and further denoting its test-test and train-test blocks respectively as ΔK_τ and $\Delta K_{t\tau}$, we finally have for the generalization error:

$$\begin{aligned} \epsilon_\tau = &\frac{1}{2}\,\mathrm{Tr}\left(\Delta K_\tau - \Delta K_{t\tau}^T\Delta\Sigma_t^{-1}\Delta K_{t\tau}\right) + \frac{1}{2}\left\|y_\tau - \Delta K_{t\tau}^T\Delta\Sigma_t^{-1}y_t\right\|^2 + \\ &\frac{1}{2}Q_{tt}\,\mathrm{Tr}\left(\tilde{M}K_{\tilde{t}\tilde{t}}\right) - \frac{1}{2}Q_{st}^2\,\mathrm{Tr}\left(\Sigma_s^{-1}K_{s\tilde{t}}\tilde{M}K_{s\tilde{t}}^T\right) + \\ &\frac{1}{2}Q_{st}^2y_s^T\Sigma_s^{-1}K_{s\tilde{t}}\tilde{M}K_{s\tilde{t}}^T\Sigma_s^{-1}y_s - Q_{st}y_s^T\Sigma_s^{-1}K_{s\tilde{t}}\tilde{M}y_{\tilde{t}} \end{aligned} \tag{70}$$

where the matrix $\tilde{M}$ reads:

$$\tilde{M} = \begin{pmatrix} \Delta\Sigma_t^{-1}\Delta K_{t\tau}\Delta K_{t\tau}^T\Delta\Sigma_t^{-1} & -\Delta\Sigma_t^{-1}\Delta K_{t\tau} \\ -\Delta K_{t\tau}^T\Delta\Sigma_t^{-1} & \mathbb{1}_{P_\tau} \end{pmatrix}. \tag{71}$$

F. Some algebraic details in our derivation

a. Useful algebraic relations. We gather here some algebraic relations involving a block-matrix $\mathcal{F}$, useful in both layer-wise integration of weights coupling and the ensuing calculations. Given four matrices $A, B, C, \Delta C$ with $A \in \mathbb{R}^{p_1 \times p_1}$, $B \in \mathbb{R}^{p_1 \times p_2}$ and $C, \Delta C \in \mathbb{R}^{p_2 \times p_2}$, we call $\mathcal{F}$ the $(p_1 + np_2) \times (p_1 + np_2)$ dimensional block matrix of the form:

$$\mathcal{F} = \begin{pmatrix} A & B & B & B \\ B^T & C + \Delta C & C & C \\ B^T & C & C + \Delta C & C \\ B^T & C & C & C + \Delta C \end{pmatrix} . \tag{72}$$

We can easily compute its inverse by first considering its lower-right $n \times n$ block

$$\mathcal{F}_n = \mathbb{1}_n \otimes \Delta C + \mathbb{I}_n \mathbb{I}_n^T \otimes C \tag{73}$$

where:

$$\mathcal{F}_n^{-1} = \mathbb{1}_n \otimes \Delta C^{-1} + \mathbb{I}_n \mathbb{I}_n^T \otimes \bar{\mathcal{F}} \tag{74}$$

$$\bar{\mathcal{F}} = \frac{1}{n} \left[(\Delta C + nC)^{-1} - \Delta C^{-1} \right] = - (\Delta C + nC)^{-1} C \Delta C^{-1} . \tag{75}$$

We have for $\mathcal{F}^{-1}$ the block structure:

$$\mathcal{F}^{-1} = \begin{pmatrix} \mathcal{F}_0^{-1} & \mathbb{I}_n \otimes \mathcal{F}_1 \\ \mathbb{I}_n^T \otimes \mathcal{F}_1^T & \mathcal{F}_{2,n}^{-1} \end{pmatrix} \tag{76}$$

with the Schur complements and off-diagonal term reading:

$$\mathcal{F}_0 = A - nB (\Delta C + nC)^{-1} B^T \tag{77}$$

$$\mathcal{F}_1 = -A^{-1} B \left(\Delta C + nC - nB^T A^{-1} B \right)^{-1} \tag{78}$$

$$\mathcal{F}_{2,n} = \mathbb{1}_n \otimes \Delta C + \mathbb{I}_n \mathbb{I}_n^T \otimes \left(C - B^T A^{-1} B \right) \tag{79}$$

$$\mathcal{F}_{2,n}^{-1} = \mathbb{1}_n \otimes \Delta C^{-1} + \mathbb{I}_n \mathbb{I}_n^T \otimes \frac{1}{n} \left[\left(\Delta C + n \bar{\mathcal{F}}_2 \right)^{-1} - \Delta C^{-1} \right] \tag{80}$$

$$\bar{\mathcal{F}}_2 = C - B^T A^{-1} B . \tag{81}$$

As for the determinant of $\mathcal{F}$ one has:

$$\log\det \mathcal{F}_n = (n-1) \log\det \Delta C + \mathrm{Tr} \log \left(\Delta C + nC \right) \tag{82}$$

$$\log\det \mathcal{F} = \log\det \mathcal{F}_n + \mathrm{Tr} \log \left(\mathcal{F}^0 \right) . \tag{83}$$

We will consider the contraction of $\mathcal{F}$ with a vector of the form $\tilde{y} = (y_0, \underbrace{y, ..., y}_{n})$:

$$\tilde{y}^T \mathcal{F}^{-1} \tilde{y} = y_0^T \mathcal{F}_0^{-1} y_0 + 2n y_0^T \mathcal{F}_1 y + n y^T \left(\Delta C + n \bar{\mathcal{F}}_2 \right)^{-1} y \tag{84}$$

Deriving the final action for transfer learning involves the $n \to 0$ limits of the previously computed quantities:

$$\log\det \mathcal{F}_n \sim n \log\det \Delta C + n \,\mathrm{Tr} \left(\Delta C^{-1} C \right) \tag{85}$$

$$\mathrm{Tr} \log \mathcal{F}^0 \sim \mathrm{Tr} \log A - n \,\mathrm{Tr} \left(A^{-1} B \Delta C^{-1} B^T \right) \tag{86}$$

$$\mathcal{F}_0^{-1} \sim A^{-1} + n A^{-1} B \Delta C^{-1} B^T A^{-1} \tag{87}$$

$$\mathcal{F}_1 \sim -A^{-1} B \left[\Delta C^{-1} - n \Delta C^{-1} \left(C - B^T A^{-1} B \right) \Delta C^{-1} \right] \tag{88}$$

$$y^T \mathcal{F}^{-1} y \sim y_0^T \mathcal{F}_0^{-1} y_0 + 2n y_0^T \mathcal{F}_1 y + n y^T \Delta C^{-1} y \tag{89}$$

b. Layer-wise integration over weights The coupling over the replicated weights involves an un-normalized Gaussian with zero and inverse covariance Λ

$$\varphi\left(w;\Lambda\right)=\prod_{ij}e^{-\sum_{a}\Lambda_{ab}w_{ij}^{a}w_{ij}^{b}} \tag{90}$$

such that integration over the replicated weights w_{ij}^{a} takes the form:

$$\int\mathcal{D}w\varphi\left(w;\Lambda\right)\prod_{ki}e^{-iw^{T}\bar{q}}=\prod_{ki}\det\Lambda e^{-\frac{1}{2}\bar{q}_{ki}^{T}\Lambda^{-1}\bar{q}_{ki}}. \tag{91}$$

In particular, we model transfer learning with an $(n+1)\times(n+1)$ matrix:

$$\Lambda=\begin{pmatrix}\lambda_{s}+\gamma n & -\gamma\mathbb{I}_{n}^{T}\\ -\gamma\mathbb{I}_{n} & \tilde{\lambda}\mathbb{1}_{n}\end{pmatrix} \tag{92}$$

with the notation $\tilde{\lambda}=\lambda_{t}+\gamma$. Calling $c=\lambda_{t}\lambda_{s}-n\gamma^{2}$, its inverse and determinant read:

$$\Lambda^{-1}=\begin{pmatrix}\frac{\tilde{\lambda}}{c} & \frac{\gamma}{c}\mathbb{I}_{n}^{T}\\ \frac{\gamma}{c}\mathbb{I}_{n} & D^{-1}\end{pmatrix} \tag{93}$$

$$\log\det\Lambda=\left(n-1\right)\log\tilde{\lambda}+\log c\sim n\log\left(\lambda_{t}+\gamma\right)+n\frac{\lambda_{t}\gamma}{\left(\lambda_{t}+\gamma\right)\lambda_{s}}+\log\lambda_{s} \tag{94}$$

with:

$$D=\tilde{\lambda}\mathbb{1}_{n}-\frac{\gamma^{2}}{\lambda_{s}+\gamma n}\mathbb{I}_{n}\mathbb{I}_{n}^{T} \tag{95}$$

$$D^{-1}=\frac{1}{\tilde{\lambda}}\left(\mathbb{1}_{n}+\frac{\gamma^{2}}{\tilde{\lambda}\lambda_{s}+n\lambda_{t}\gamma}\mathbb{I}_{n}\mathbb{I}_{n}^{T}\right) \tag{96}$$

c. Details on determinants and quadratic forms The determinant of the $(n+1)\times(n+1)$ matrix $\det\left(\mathbb{1}_{n+1}+\bar{\mathcal{Q}}\right)$ can be easily worked out by using formulas from the previous section:

$$\begin{aligned}&\det\left(\mathbb{1}_{n+1}+\bar{\mathcal{Q}}\right)=\det\bar{\mathcal{Q}}_{n}\left(1+\bar{Q}_{s}-\bar{Q}_{st}^{2}\mathbb{I}_{n}^{T}\bar{\mathcal{Q}}_{n}^{-1}\mathbb{I}_{n}\right)=\\&\left(1+\bar{Q}_{t}-\bar{Q}_{tt}\right)^{n-1}\left(1+\bar{Q}_{t}-\bar{Q}_{tt}+n\bar{Q}_{tt}\right)\left(1+\bar{Q}_{s}-\frac{n\bar{Q}_{st}^{2}}{1+\bar{Q}_{t}-\bar{Q}_{tt}+n\bar{Q}_{tt}}\right)\end{aligned} \tag{97}$$

In the small n limit it has the form:

$$\begin{aligned}&\log\det\left(\mathbb{1}_{n+1}+\bar{\mathcal{Q}}\right)=\log\left(1+\bar{Q}_{s}\right)+\\&n\left[\log\left(1+\bar{Q}_{t}-\bar{Q}_{tt}\right)+\frac{\left(1+\bar{Q}_{s}\right)\bar{Q}_{tt}-\bar{Q}_{st}^{2}}{\left(1+\bar{Q}_{s}\right)\left(1+\bar{Q}_{t}-\bar{Q}_{tt}\right)}\right].\end{aligned} \tag{98}$$

We recall that the replicated coupling matrix for last-layer pre-activations reads:

$$\Sigma=\begin{pmatrix}\tilde{\Sigma}_{s} & Q_{st}K_{st} & Q_{st}K_{st} & ... & Q_{st}K_{st}\\ Q_{st}K_{st}^{T} & \Sigma_{t} & Q_{tt}K_{tt} & ... & Q_{tt}K_{tt}\\ Q_{st}K_{st}^{T} & Q_{tt}K_{tt} & \Sigma_{t} & ... & ...\\ ... & ... & ... & ... & Q_{tt}K_{tt}\\ Q_{st}K_{st}^{T} & Q_{tt}K_{tt} & ... & Q_{tt}K_{tt} & \Sigma_{t}\end{pmatrix} \tag{99}$$

with $\tilde{\Sigma}_{s}=\frac{\mathbb{1}}{\beta_{s}}+Q_{s}K_{s}$. In the expansion of $\log\det\Sigma$ and the quadratic form $y^{T}\Sigma^{-1}y$, we encounter the terms

$$\tilde{\Sigma}_{s}=\left(\frac{\mathbb{1}}{\beta_{s}}+Q_{s}\left(K_{s}+n\delta K_{s}\right)\right)^{-1}\sim\Sigma_{s}^{-1}-nQ_{s}\Sigma_{s}^{-2}\delta K_{s} \tag{100}$$

arising from $\tilde{K}_{s,\mu\nu}=\left\langle\sigma\left(h_{\mu}\right)\sigma\left(h_{\nu}\right)\right\rangle\sim K_{s,\mu\nu}+n\delta K_{s,\mu\nu}$. We will however drop the $\mathcal{O}\left(n\right)$ terms since they do not contribute to the SP equations for the transfer order parameters. We thus have for the two quantities:

$$\begin{aligned}&\log\det\left(B^{-1}+\Sigma\right)\sim\mathrm{Tr}\log\Sigma_{s}+n\log\det\Delta\Sigma_{t}+nQ_{tt}\,\mathrm{Tr}\left(\Delta\Sigma_{t}^{-1}K_{tt}\right)\\&-nQ_{st}^{2}\,\mathrm{Tr}\left(\Sigma_{s}^{-1}K_{st}\Delta\Sigma_{t}^{-1}K_{st}^{T}\right)\end{aligned} \tag{101}$$

$$\begin{aligned}&y^{T}\left(B^{-1}+\Sigma\right)^{-1}y\sim y_{0}^{T}\Sigma_{s}^{-1}y_{0}+nQ_{st}^{2}y_{0}^{T}\Sigma_{s}^{-1}K_{st}\Delta\Sigma_{t}^{-1}K_{st}^{T}\Sigma_{s}^{-1}y_{0}+\\&-2nQ_{st}y_{0}^{T}\Sigma_{s}^{-1}K_{st}\Delta\Sigma_{t}^{-1}y+ny^{T}\Delta\Sigma_{t}^{-1}y\,.\end{aligned} \tag{102}$$

III. TRANSFER LEARNING IN A SIMPLE REGRESSION PROBLEM

In this short supplementary section, we address the transfer learning problem in the simplest possible setting, linear regression for both the *source* and the *target* task.

A. Derivation

The learning problem for the source vector w_s is defined by the minimization of the regularized training loss:

$$\mathcal{L}_s = \frac{1}{2}\sum_{\mu=1}^{P_s}\left(y_s^\mu - \sum_i \frac{w_{s,i}x_{s,i}^\mu}{\sqrt{N}}\right)^2 + \frac{N\lambda_s}{2}\sum_{i=1}^{N} w_{s,i}^2 \tag{103}$$

which yields the known solution:

$$w_s = \left(X_s^T X_s + N\lambda_s \mathbb{1}_N\right)^{-1} X_s^T y_s\,. \tag{104}$$

With the help of the Woodbury identity we get from the previous expression:

$$w_s = \frac{1}{N} X_s^T G_s y_s \tag{105}$$

with the definition

$$G_s = (\lambda_s \mathbb{1}_{P_s} + C_s)^{-1} \Leftrightarrow \mathbb{1}_{P_s} - G_s C_s = \lambda_s G_s. \tag{106}$$

The transfer learning problem for the target weight vector w_t is defined by the minimization of the regularized training loss in the presence of a source-target coupling:

$$\mathcal{L}_t = \frac{1}{2}\sum_{\mu=1}^{P_t}\left(y_t^\mu - \sum_i \frac{w_{t,i}x_{t,i}^\mu}{\sqrt{N}}\right)^2 + \frac{N\lambda_t}{2}\sum_{i=1}^{N} w_{t,i}^2 + \frac{N\gamma}{2}\sum_{i=1}^{N}\left(w_{t,i} - w_{s,i}\right)^2 \tag{107}$$

yielding:

$$w = \left(X_t^T X_t + N\tilde{\lambda}\mathbb{1}_N\right)^{-1}\left(X^T y_t + N\gamma w_s\right) \tag{108}$$

with the notation $\tilde{\lambda} = \lambda_t + \gamma$. Again, by calling $G_t = \left(\tilde{\lambda}\mathbb{1}_{P_t} + C_t\right)^{-1}$, we have:

$$\left(\frac{X_t^T X_t}{N} + \tilde{\lambda}\mathbb{1}_N\right)^{-1} = \frac{1}{\tilde{\lambda}}\left(\mathbb{1}_N - \frac{1}{N}X_t^T G_t X_t\right)\,. \tag{109}$$

Introducing the notations

$$\tilde{y}_s = \lambda_s G_s y_s \tag{110}$$

$$\tilde{y}_t = \tilde{\lambda} G_t y_t \tag{111}$$

$$\tilde{Y} = [\tilde{y}_s, \tilde{y}_t] \tag{112}$$

we write succintly:

$$w_t = \frac{1}{N\tilde{\lambda}}\begin{pmatrix} \frac{\gamma}{\lambda_s}\left(X_s^T - X_t^T G_t C_{st}^T\right) \\ X_t^T \end{pmatrix}\tilde{Y} \tag{113}$$

where we have defined $C_{st} = \frac{X_s X_t^T}{N}$.

a. Norm of target weight vector The norm of w_t is easily written as:

$$\|w_t\|^2 = \frac{1}{N\tilde{\lambda}^2}\tilde{Y}\tilde{\mathcal{W}}\tilde{Y}^T \tag{114}$$

defining the matrix:

$$\tilde{\mathcal{W}} = \begin{pmatrix} \tilde{\mathcal{W}}_s & \tilde{\mathcal{W}}_{st} \\ \tilde{\mathcal{W}}_{st}^T & \tilde{\mathcal{W}}_t \end{pmatrix} \tag{115}$$

$$\tilde{\mathcal{W}}_s = \frac{\gamma^2}{\lambda_s^2}\left[C_s - C_{st}G_t\left(\mathbb{1}_{P_t} + \tilde{\lambda}G_t\right)C_{st}^T\right] \tag{116}$$

$$\tilde{\mathcal{W}}_{st} = \frac{\gamma\tilde{\lambda}}{\lambda_s}C_{st}G_t \tag{117}$$

$$\tilde{\mathcal{W}}_t = C_t\,. \tag{118}$$

b. Training and test error Inserting the solution w_t in the expression for the training error we get, after some manipulation:

$$\epsilon_t = \frac{1}{2}\|y_t - X_t w_t\|^2 = \frac{1}{2}\frac{\gamma^2}{\lambda_s^2}\tilde{y}_s^T C_{st}G_t^2 C_{st}^T\tilde{y}_s - \frac{\gamma}{\lambda_s}\tilde{y}_s^T C_{st}G_t\tilde{y}_t + \frac{1}{2}\tilde{y}_t^T\tilde{y}_t\,. \tag{119}$$

The error over the test set is given by the following:

$$\begin{aligned}\epsilon_g = &\frac{1}{2}\|y_\tau - X_\tau w_t\|^2 = \\ &\frac{\gamma^2}{2\lambda_s^2\tilde{\lambda}^2}\tilde{y}_s^T G_{s\tau}G_{s\tau}^T\tilde{y}_s + \\ &\frac{1}{2\tilde{\lambda}^2}\tilde{y}^T C_{t\tau}C_{t\tau}^T\tilde{y}_t + \frac{\gamma}{\lambda_s\tilde{\lambda}^2}\tilde{y}_s^T G_{s\tau}C_{t\tau}^T\tilde{y}_t \\ &-\frac{\gamma}{\lambda_s\tilde{\lambda}}y_\tau^T G_{s\tau}^T\tilde{y}_s - \frac{1}{\tilde{\lambda}}y_\tau^T C_{t\tau}^T\tilde{y}_t + \frac{1}{2}y_\tau^T y_\tau\end{aligned} \tag{120}$$

with the source-test and target-test covariances defined respectively as $C_{s\tau} = \frac{X_s X_\tau^T}{N}$ and $C_{t\tau} = \frac{X_t X_\tau^T}{N}$ and $G_{s\tau} = C_{s\tau} - C_{st}G_t C_{t\tau}$.

B. A simple example of transfer regression

In Figure 1 we show an example of a transfer learning problem in a single-layer, linear version of the task shown in Fig. 2 of the main text, in the special case with two identical teachers for the source and target tasks. We generate the three datasets X_s, X_t and X_t with normal i.i.d. entries. The task is defined in terms of a random teacher weight vector w_0 with zero-mean and Gaussian i.i.d. entries with standard deviation $1/N$ (we use $N = 200$ in this examples): outputs for the three sets are linear functions of the inputs X_α given by $y_\alpha = X_\alpha w_0$ for $\alpha \in \{s, t, \tau\}$. The source weight vector w_s is trained using $P_s = \alpha_s N$ data-points, with $\alpha_s = 0.8$. The transfer effect is apparent in the decrease of the generalization error as a function of the source-target coupling parameter γ.

IV. TRANSFER LEARNING IN A DEEP FULLY CONNECTED NETWORK

In this section, we sketch a tentative derivation of the effective action in the case of networks with L hidden layers. In this case, all the layers except for the last are coupled among source and target, while the readout ($L + 1$ in our convention) remains uncoupled. The replicated priors for the weights at each layer are defined by the coupling matrices Λ_l:

$$\Lambda_l = \begin{pmatrix} \lambda_{s,l} & -\gamma & -\gamma & ... & -\gamma \\ -\gamma & \lambda_{t,l} & 0 & ... & 0 \\ -\gamma & 0 & \lambda_{t,l} & ... & ... \\ ... & ... & ... & ... & 0 \\ -\gamma & 0 & ... & 0 & \lambda_{t,l} \end{pmatrix}, \qquad \Lambda_{L+1} = \begin{pmatrix} \lambda_{s,L+1} & 0 & 0 & ... & 0 \\ 0 & \lambda_{t,L+1} & 0 & ... & 0 \\ 0 & 0 & \lambda_{t,L+1} & ... & ... \\ ... & ... & ... & ... & 0 \\ 0 & 0 & ... & 0 & \lambda_{t,L+1} \end{pmatrix}. \tag{121}$$

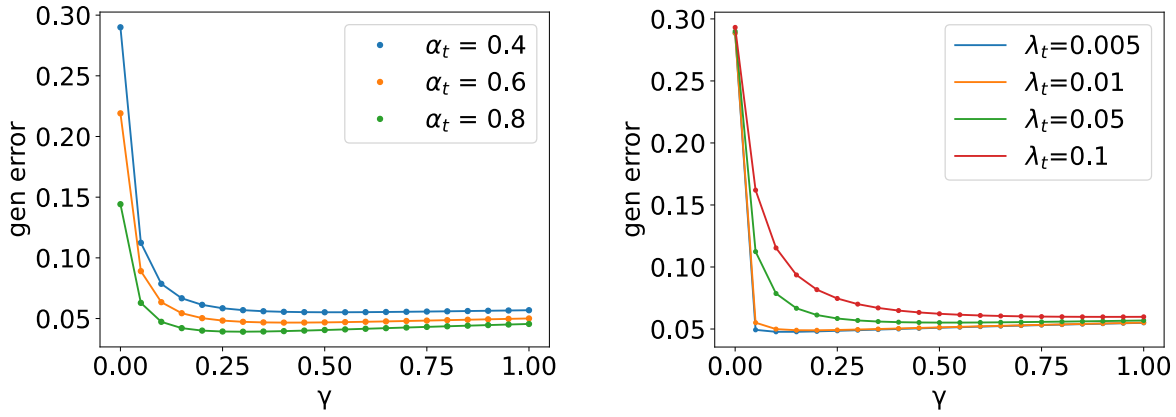

FIG. 1. Transfer learning in a linear regression task defined by a random teacher vector w_0. In both panels curves are theoretical results, points are obtained by solving the source and target problem with standard optimization methods. Parameters: $N = 200$, $P_\tau = 10.000$ **A**: Generalization error as a function of the coupling parameter γ for different sizes of the training set for the target problem, indicated by the ratio $\alpha_t = P_t/N$. **B**: Same as **A** for different values of the regularization λ_t of the target weight vector w_t.

We are interested in the computation of the replicated partition function Z^n, using index $a = 1 \dots n$ for replicas. To describe a fully connected layer with weight matrix w_l, we introduce the following notation for the pre-activations at layer $l+1$:

$$h^a_{l+1} = w^a_l x^a_l, \qquad x^a_l = \sigma\left(h^a_l\right) . \tag{122}$$

To easily deal with the recursion over layer, let us introduce a definition for the function computed by a neural network between any two intermediate layers l, l':

$$h^a_{l'} = \phi^a_{l:l'}\left(h_l; \{w^a\}_{l+1:l'}\right) . \tag{123}$$

The function $\phi_{l:l'}$ takes as inputs the pre-activations at layer l and outputs the pre-activations at layer l', with the convention that $h_0 = x$. We have stressed the dependence of $\phi_{l:l'}$ on all the weights of the layers between l and l', $\{w\}_{l+1:l'}$, but we will immediately drop it to ease the notation. The output of the network reads $\phi\left(x\right) \equiv \phi_{0:L+1}\left(x\right)$, and we can write the loss in terms of the pre-activations of layer l as:

$$\mathcal{L}\left(\{w\}\right) = \frac{1}{2}\sum_\mu \left(\phi\left(x^\mu\right) - y^\mu\right)^2 = \sum_\mu \left(\phi_{l:L+1}\left(h^\mu_l\right) - y^\mu\right)^2 . \tag{124}$$

Let us define:

$$\chi_{l:l'}\left(h^\mu_l; \{w\}_{l+1:l'}\right) = \exp\left[-\frac{1}{2}\sum_{\mu a}\beta_a\left(\phi_{l:l'}\left(h^\mu_l\right) - y^a_\mu\right)^2\right] ,$$

Using this convention we can rewrite the free-entropy as:

$$f = \frac{1}{N_L}\frac{1}{Z_s\left(\beta_s\right)}\lim_{n\to 0}\partial_n Z^n \tag{125}$$

$$Z^n = \int \prod_l \mathcal{D}w_l\, \chi_{0:L+1}(x) e^{-\frac{1}{2}\sum_l w_l^T \Lambda_l w_l} , \tag{126}$$

with the shorthands $\mathcal{D}w_l = \prod_{i_l i_{l-1},a} dw^{l,a}_{i_l i_{l-1}}$ and $w_l^T\Lambda_l w_l = \sum_{i_l i_{l-1}}\sum_{ab} w^{l,a}_{l,i_l i_{l-1}}\Lambda_{l,ab} w^{l,b}_{l,i_l i_{l-1}}$.

A. Integrating first-layer weights

The integration over the first-layer weights $w^a_{1,i_0 i_1}$ is straightforward, since for each i_1, i_0 the coupling is Gaussian with inverse covariance Λ_1. Let us isolate the integral over the first-layer weights and consider the dependence over

the first layer pre-activations:

$$Z^n = \int \prod_{l>1} \mathcal{D}w_l e^{-\frac{1}{2}\sum_{l>1} w_l^T \Lambda_l w_l} \int \mathcal{D}h_1 \chi_{1:L+1}(h_1)\, \psi(h_1) \tag{127}$$

where:

$$\psi(h_1) = \prod_{i_1} \int \prod_{\mu} d\bar{h}^a_{1,\mu i_1} e^{i\sum_{a\mu} \bar{h}^a_{1,\mu i_1} h^a_{1,\mu i_1}} \prod_{i_0} \int dw e^{-\frac{1}{2}\sum_{ab} w^a \Lambda_1^{ab} w^b - \frac{i}{\sqrt{N_0}} \sum_{a\mu} w^a \bar{h}^a_{1,\mu i_1} x^a_{\mu i_0}}. \tag{128}$$

Integrating over the weights and using the notation $\Delta_l = \exp\{-\frac{1}{2} N_{l-1} N_l \log\det \Lambda_l\}$, we have:

$$\psi(h_1) = \Delta_1 \prod_{i_1 i_0} \int \prod_{\mu} d\bar{h}^a_{1,\mu i_1} e^{i\sum_{a\mu} \bar{h}^a_{1,\mu i_1} \bar{h}^a_{1,\mu i_1} - \frac{1}{2}\sum_{ab} \Lambda^{-1}_{ab} \bar{q}^a_{0,i_1 i_0}(\bar{h}) \bar{q}^b_{0,i_1 i_0}(\bar{h})} \tag{129}$$

with the definition

$$\bar{q}^a_{0,i_1 i_0}(\bar{h}) = \frac{1}{\sqrt{N_0}} \sum_{\mu} \bar{h}^a_{\mu i_1} x^a_{\mu i_0}\,. \tag{130}$$

Employing the factorization over the first layer index i_1 and summing over i_0, we can write the replicated partition function as follows

$$Z^n = \Delta_1 \int \prod_{l>1} \mathcal{D}w_l e^{-\frac{1}{2}\sum_{l>1} w_l^T \Lambda_l w_l} \int \prod_{i_1} \{\mathcal{D}h_{1,i_1} \mathcal{N}(h_{1,i_1}; 0, \Sigma_1)\}\, \chi_{1:L+1}\left(\{h_1\}_{i_1}\right) \tag{131}$$

where, for each i_1, the replicated pre-activations are Gaussian with covariance matrices Σ_1:

$$\tilde{C}^{ab}_{\mu\nu} = \left\langle h^a_{1,\mu i_1} h^b_{1,\nu i_1} \right\rangle = \left(\Lambda_1^{-1}\right)^{ab} C^{ab}_{\mu\nu} \tag{132}$$

in turn depending on the replicated input covariances:

$$C^{ab}_{\mu\nu} = \frac{1}{N_0} \sum_{i_0=1}^{N_0} x^a_{\mu i_0} x^b_{\nu i_0}\,. \tag{133}$$

B. Recursion relation for deep FC network

Introducing the definition for the second-layer pre-activations we have:

$$Z^n = \Delta_1 \int \prod_{l>2} \mathcal{D}w_l e^{-\frac{1}{2}\sum_{l>2} w_l^T \Lambda_l w_l} \int \mathcal{D}h_2 \chi_{2:L+1}(h_2)\, \psi(h_2) \tag{134}$$

with:

$$\begin{aligned}\psi(h_2) = \int \mathcal{D}\bar{h}_2 e^{i\bar{h}_2^T h_2} \int \prod_{i_1} \{\mathcal{D}h_{1,i_1} \mathcal{N}(h_{1,i_1}; 0, \Sigma_1)\} \\ \int \mathcal{D}w_2 \prod_{i_2 i_1} e^{-\frac{i}{\sqrt{N_1}} \sum_a w^a_{2,i_2 i_1} \sum_\mu \bar{h}^a_{2,\mu i_2} \sigma\left(h^a_{1,\mu i_1}\right)}\end{aligned} \tag{135}$$

We again introduce the quantities

$$\bar{q}^a_{1,i_2 i_1}(\bar{h}_2) = \frac{1}{\sqrt{N_1}} \sum_{\mu} \bar{h}^a_{2,\mu i_2} \sigma\left(h^a_{1,\mu i_1}\right) \tag{136}$$

and perform the integration over the weights w_2. Employing the factorization over the index i_1 (and dropping the index for clarity) we write

$$\psi(h_2) = \Delta_2 \prod_{i_2 i_1} \int \mathcal{D}\bar{h}_2 e^{i\bar{h}_2^T h_2} \left(\int \mathcal{D}\bar{q}_1 \varphi(\bar{q}_1)\, e^{-\frac{1}{2}\bar{q}_1^T \bar{q}_1 - \frac{1}{2}\bar{q}_1^T \Lambda_2^{-1} \bar{q}_1} \right)^{N_1} \tag{137}$$

with:

$$\varphi\left(\bar{q}_1\right)=\left\langle\prod_{i_2}\delta\left(\bar{q}_{1,i_2}^a-\frac{1}{\sqrt{N_1}}\sum_\mu \bar{h}_{2,\mu i_2}^a\sigma\left(h_{1,\mu}^a\right)\right)\right\rangle_{h_1} \tag{138}$$

To deal with the extensive extensive $N_2 n$ number of variables $\bar{q}_1$, we employ a self-consistent Gaussian approximation with covariance matrix

$$\bar{\mathcal{Q}}_{1,i_2 j_2}^{ab}=\left\langle\bar{q}_{1,i_2}^a\bar{q}_{1,j_2}^b\right\rangle=\frac{1}{N_1}\sum_{\mu\nu}\bar{h}_{2,\mu i_2}^a K_{1,\mu\nu}^{ab}\bar{h}_{2,\nu j_2}^b \tag{139}$$

and kernels

$$K_{1,\mu\nu}^{ab}=\left\langle\sigma\left(h_{1,\mu}^a\right)\sigma\left(h_{1,\nu}^{ab}\right)\right\rangle_{\mathcal{N}\left(h_1;0,\tilde{C}\right)} \tag{140}$$

thus getting:

$$\begin{aligned}\left\langle e^{-\frac{1}{2}\bar{q}_1^T\Lambda_2^{-1}\bar{q}_1}\right\rangle_{\varphi(\bar{q}_1)}&=e^{-\frac{N_1}{2}\operatorname{Tr}_{i_2,a}\log\left(\mathbb{1}+\left(\Lambda_2^{-1}\right)^{ab}\bar{\mathcal{Q}}_{1,i_2 j_2}^{ab}\right)}\sim\\&e^{-\frac{N_1}{2}\operatorname{Tr}_a\log\left(\mathbb{1}+\left(\Lambda_2^{-1}\right)^{ab}\operatorname{Tr}_{i_2}\bar{\mathcal{Q}}_{1,i_2 j_2}^{ab}\right)}\end{aligned} \tag{141}$$

The second line of the previous equations implement a mean-field, permutation symmetric approximation, whereby we obtained an inter-replica covariance by tracing over the N_2 second-layer hidden units.

Introducing the definitions for the new mean-field inter-replica covariance with appropriate δ functions, we thus get:

$$\begin{aligned}\psi\left(h_2\right)=\Delta_2\int\mathcal{D}\bar{\mathcal{Q}}_1 e^{-\frac{1}{2}\log\det\left(\mathbb{1}+\Lambda_2^{-1}\bar{\mathcal{Q}}_1\right)}\\\int\mathcal{D}\bar{h}e^{i\bar{h}^T h_2}\delta\left(N_1\bar{\mathcal{Q}}_1^{ab}-\left(\bar{h}^a\right)^T K_1^{ab}\bar{h}^b\right)\end{aligned} \tag{142}$$

Expanding the δ's and easily integrating over $\bar{h}$ we find:

$$\psi\left(h_2\right)=\Delta_2\int\mathcal{D}\mathcal{Q}_1\mathcal{D}\bar{\mathcal{Q}}_1 e^{\frac{N_1}{2}\operatorname{Tr}\left(\mathcal{Q}_1\bar{\mathcal{Q}}_1\right)-\frac{1}{2}\log\det\left(\mathbb{1}+\Lambda_2^{-1}\bar{\mathcal{Q}}_1\right)}\mathcal{N}\left(h_2;0,\mathcal{K}_2\right) \tag{143}$$

with

$$\mathcal{K}_2^{ab}=\mathcal{Q}_1^{ab}K_1^{ab} \tag{144}$$

The matrix $\mathcal{Q}_1$ acts as renormalization for the kernel K_1, whereby h_2 are Gaussian conditioning on $\mathcal{Q}_1$. Using a simple recursion across layer one gets:

$$\begin{aligned}Z^n=\prod_{l=1}^L\Delta_l\int\prod_{l=1}^{L-1}\mathcal{D}\mathcal{Q}_l\mathcal{D}\bar{\mathcal{Q}}_l e^{\frac{1}{2}\sum_{l=1}^{L-1}N_l\operatorname{Tr}\left(\mathcal{Q}_l\bar{\mathcal{Q}}_l\right)-\frac{1}{2}\sum_{l=1}^{L-1}\log\det\left(\mathbb{1}+\Lambda_{l+1}^{-1}\bar{\mathcal{Q}}_l\right)}\\\int\mathcal{D}v e^{-\frac{1}{2}v^T\Lambda_{L+1}v}\int\mathcal{D}h^L\chi_{L:L+1}\left(h^L\right)\psi\left(h^L\right)\end{aligned} \tag{145}$$

with the notation $v\equiv w_{L+1}$ on the uncoupled last layer weights.

C. Readout layer

Introducing the definition $s=\phi_{L:L+1}\left(h_L;v\right)$ for the readout outputs, we obtain a form

$$Z^n=\prod_{l=1}^L\Delta_l\int\prod_{l=1}^{L-1}\mathcal{D}\mathcal{Q}_l\mathcal{D}\bar{\mathcal{Q}}_l e^{\frac{1}{2}\sum_{l=1}^{L-1}N_l\operatorname{Tr}\left(\mathcal{Q}_l\bar{\mathcal{Q}}_l\right)-\frac{1}{2}\sum_{l=1}^{L-1}\log\det\left(\mathbb{1}+\Lambda_{l+1}^{-1}\bar{\mathcal{Q}}_l\right)}\int\mathcal{D}s\psi\left(s\right) \tag{146}$$

with

$$\psi\left(s\right)=\Delta_{L+1}\int\mathcal{D}\bar{s}e^{i\bar{s}^{T}s-\frac{1}{2}\left(s-y\right)^{T}B\left(s-y\right)}\int\mathcal{D}h_{L}\mathcal{N}\left(h_{L};0,\mathcal{K}_{L}\right)$$
$$\int\mathcal{D}v\prod_{i_{L}}e^{-\frac{i}{\sqrt{N_{L}}}\sum_{a}v_{i_{L}}^{a}\sum_{\mu}\bar{s}_{\mu}^{a}\sigma\left(h_{L,\mu i_{L}}^{a}\right)} \tag{147}$$

We again introduce the variables $\bar{q}^{a}=\frac{1}{\sqrt{N_{L}}}\sum_{\mu}\bar{s}_{\mu}^{a}\sigma\left(h_{L,\mu}^{a}\right)$, where we dropped the index i_L owning to factorization. Employing the usual Gaussian equivalence, their joint distribution is a normal distribution with order-parameter covariance matrix

$$\bar{\mathcal{Q}}_{L}^{ab}=\left\langle \bar{q}^{a}\bar{q}^{b}\right\rangle =\frac{1}{N_{L}}\left(\bar{s}^{a}\right)^{T}K_{L}^{ab}\bar{s}^{b} \tag{148}$$

and kernels

$$K_{L,\mu\nu}^{ab}=\left\langle \sigma\left(h_{\mu}^{a}\right)\sigma\left(h_{\nu}^{a}\right)\right\rangle _{\mathcal{N}\left(h_{L};0,\mathcal{K}_{L}\right)} \tag{149}$$

We finally obtain

$$\psi\left(s\right)=\Delta_{L+1}\int\mathcal{D}\mathcal{Q}_{L}\mathcal{D}\bar{\mathcal{Q}}_{L}e^{\frac{N_{L}}{2}\mathrm{Tr}\left(\mathcal{Q}_{L}\bar{\mathcal{Q}}_{L}\right)-\frac{1}{2}\log\det\left(\mathbb{1}+\Lambda_{L+1}^{-1}\bar{\mathcal{Q}}_{L}\right)}$$
$$\int\mathcal{D}\bar{s}e^{i\bar{s}^{T}s-\frac{1}{2}\bar{s}^{T}\mathcal{K}_{L+1}\bar{s}-\frac{1}{2}\left(s-y\right)^{T}B\left(s-y\right)} \tag{150}$$

where $\mathcal{K}_{L+1}^{ab}=\mathcal{Q}_{L}^{ab}K_{L}^{ab}$. We now employ the same steps as in the case of the 1hl network, thus arriving at the final form

$$Z^{n}=\int\prod_{l}\mathcal{D}\mathcal{Q}_{l}\mathcal{D}\bar{\mathcal{Q}}_{l}e^{\frac{N_{L}}{2}S\left(\mathcal{Q},\bar{\mathcal{Q}}\right)} \tag{151}$$

with the action

$$S=\sum_{l}^{L}\left[\log\Delta_{l}+\frac{N_{l}}{N_{L}}\mathrm{Tr}\left(\mathcal{Q}_{l}\bar{\mathcal{Q}}_{l}\right)-\frac{N_{l}}{N_{L}}\sum_{l}\log\det\left(\mathbb{1}+\Lambda_{l+1}^{-1}\bar{\mathcal{Q}}_{l}\right)\right]+$$
$$\log\Delta_{L+1}-\log\det\left(\mathbb{1}+B\mathcal{K}_{L+1}\right)-y^{T}\left(B^{-1}+\mathcal{K}_{L+1}\right)^{-1}y\,. \tag{152}$$

The $n\to0$ limit can again be carried out using the expressions for the determinants and quadratic forms in section II F 0 a. It is worth noticing that we expect this derivation to be exact for deep linear networks, as long as replica symmetry is not broken.

---

[1] R. Pacelli, S. Ariosto, M. Pastore, F. Ginelli, M. Gherardi, and P. Rotondo, A statistical mechanics framework for bayesian deep neural networks beyond the infinite-width limit, Nature Machine Intelligence **5**, 1497 (2023).

[2] R. Aiudi, R. Pacelli, A. Vezzani, R. Burioni, and P. Rotondo, Local Kernel Renormalization as a mechanism for feature learning in overparametrized Convolutional Neural Networks, arXiv e-prints , arXiv:2307.11807 (2023), arXiv:2307.11807 [cs.LG].

[3] C. Williams, Computing with infinite networks, in *Advances in Neural Information Processing Systems*, Vol. 9, edited by M. Mozer, M. Jordan, and T. Petsche (MIT Press, 1996).