A Deferred proofs

A.1 Alternate characterizations of GULP, proofs of Proposition 1 and Lemma 1

We provide proofs of the two alternative characterizations to the GULP distance that were claimed in the main text.

Proof of Lemma 1. Fix a distribution of (X, Y), and let $\eta(x) = \mathbb{E}[Y|X = x]$ be the regression function. Since we are using squared error, with ϕ features the best linear predictor is β_λ that solves

$$\beta_\lambda = \Sigma_\phi^{-\lambda}\,\mathbb{E}[Y\phi(X)] = \Sigma_\phi^{-\lambda}\,\mathbb{E}[\eta(X)\phi(X)] = \Sigma_\phi^{-\lambda}\int \eta(x)\phi(x)\mathrm{d}P_X(x)$$

where P_X is the marginal distribution of X. Similarly

$$\gamma_\lambda = \Sigma_\psi^{-\lambda}\int \eta(x)\psi(x)\mathrm{d}P_X(x)$$

In particular, for a given distribution of (X, Y), the distance between the best linear predictors is

$$\mathbb{E}(\beta_\lambda^\top\phi(X) - \gamma_\lambda^\top\psi(X))^2.$$

We rewrite this in terms of η:

$$\begin{aligned}
\mathbb{E}(\beta_\lambda^\top\phi(X) - \gamma_\lambda^\top\psi(X))^2 &= \mathbb{E}\left(\int \eta(x)\left[\phi(X)^\top\Sigma_\phi^{-\lambda}\phi(x) - \psi(X)^\top\Sigma_\psi^{-\lambda}\psi(x)\right]\mathrm{d}P_X(x)\right)^2\\
&= \mathbb{E}\langle \eta, \phi(X)^\top\Sigma_\phi^{-\lambda}\phi(\cdot) - \psi(X)^\top\Sigma_\psi^{-\lambda}\psi(\cdot)\rangle_{L^2(P_X)}^2
\end{aligned}$$

Therefore to sup out the distribution over Y, we take the sup of η such that $\|\eta\|_{L^2(P_X)} \le 1$. It yields the claim of Lemma 1.

$$\begin{aligned}
d_\lambda^2(\phi, \psi) &:= \sup_{\|\eta\|_{L^2(P_X)}\le 1} \mathbb{E}\langle \eta, \cdots\rangle_{L^2(P_X)}^2\\
&= \mathbb{E}\,\|\phi(X)^\top\Sigma_\phi^{-\lambda}\phi(\cdot) - \psi(X)^\top\Sigma_\psi^{-\lambda}\psi(\cdot)\|_{L^2(P_X)}^2\\
&= \mathbb{E}(\phi(X)^\top\Sigma_\phi^{-\lambda}\phi(X') - \psi(X)^\top\Sigma_\psi^{-\lambda}\psi(X'))^2
\end{aligned}$$

where $X, X' \sim P_X$ are independent. □

Using Lemma 1, we can easily prove Proposition 1.

Proof of Proposition 1. Start with the characterization in Lemma 1, expand the square and use the cylicity and linearity of the trace:

$$\begin{aligned}
d_\lambda^2(\phi, \psi) &= \mathbb{E}(\phi(X)^\top\Sigma_\phi^{-\lambda}\phi(X')\phi(X')^\top\Sigma_\phi^{-\lambda}\phi(X))\\
&\quad + \mathbb{E}(\psi(X)^\top\Sigma_\psi^{-\lambda}\psi(X')\psi(X')^\top\Sigma_\psi^{-\lambda}\psi(X))\\
&\quad - 2\,\mathbb{E}(\phi(X)^\top\Sigma_\phi^{-\lambda}\phi(X')\psi(X')^\top\Sigma_\psi^{-\lambda}\psi(X))\\
&= \operatorname{tr}\mathbb{E}(\Sigma_\phi^{-\lambda}\phi(X')\phi(X')^\top\Sigma_\phi^{-\lambda}\phi(X)\phi(X)^\top)\\
&\quad + \operatorname{tr}\mathbb{E}(\Sigma_\psi^{-\lambda}\psi(X')\psi(X')^\top\Sigma_\psi^{-\lambda}\psi(X)\psi(X)^\top)\\
&\quad - 2\operatorname{tr}\mathbb{E}(\Sigma_\phi^{-\lambda}\phi(X')\psi(X')^\top\Sigma_\psi^{-\lambda}\psi(X)\phi(X)^\top)\\
&= \operatorname{tr}(\Sigma_\phi^{-\lambda}\Sigma_\phi\Sigma_\phi^{-\lambda}\Sigma_\phi) + \operatorname{tr}(\Sigma_\psi^{-\lambda}\Sigma_\psi\Sigma_\psi^{-\lambda}\Sigma_\psi) - 2\operatorname{tr}(\Sigma_\phi^{-\lambda}\Sigma_{\phi\psi}\Sigma_\psi^{-\lambda}\Sigma_{\phi\psi}^\top).
\end{aligned}$$

□

A.2 GULP is a distance, proof of Theorem 2

We complete the proof of Theorem 2 by characterizing when the GULP distance is zero in the following lemma.

Lemma 2 (Characterization for when GULP is zero, for $\lambda > 0$)**.** *For any $\lambda > 0$, the two representation maps $\phi : \mathbb{R}^d \to \mathbb{R}^k, \psi : \mathbb{R}^d \to \mathbb{R}^l$ have zero GULP distance, $d_\lambda(\phi, \psi) = 0$, if and only if $k = l$ andthere exists an orthogonal transformation $U \in \mathbb{R}$ such that $\phi(X) = U\psi(X)$ a.s.*

Proof of Lemma 2. In the main text it was shown that if ϕ and ψ are related by an orthogonal transformation, then $d_\lambda(\phi, \psi) = 0$. It remains to prove the converse direction, which is more involved. Define $\tilde{\phi}(x) = (\Sigma_\phi + \lambda I)^{-1/2}\phi(x)$ and $\tilde{\psi}(x) = (\Sigma_\psi + \lambda I)^{-1/2}\psi(x)$. We make the following claim, whose proof we defer:

Claim 1. *Let $\lambda > 0$ and suppose $d_\lambda(\phi, \psi) = 0$. Then $k = l$ and there is an orthogonal transformation $U \in \mathbb{R}^{k \times k}$ such that $\tilde{\phi}(X) = U\tilde{\psi}(X)$ almost surely.*

Let $U \in \mathbb{R}^{k \times k}$ be the orthogonal transformation guaranteed by the above claim. We can write

$$\begin{aligned}\Sigma_\phi &= \mathbb{E}[\phi(X)\phi(X)^\top] \\ &= (\Sigma_\phi + \lambda I)^{1/2}U(\Sigma_\psi + \lambda I)^{-1/2}\,\mathbb{E}[\psi(X)\psi(X)^\top](\Sigma_\psi + \lambda I)^{-1/2}U^T(\Sigma_\phi + \lambda I)^{1/2} \\ &= (\Sigma_\phi + \lambda I)^{1/2}U(\Sigma_\psi + \lambda I)^{-1/2}\Sigma_\psi(\Sigma_\psi + \lambda I)^{-1/2}U^T(\Sigma_\phi + \lambda I)^{1/2}.\end{aligned}$$

Since Σ_ϕ and $(\Sigma_\phi + \lambda I)^{1/2}$ commute, and similarly for Σ_ψ and $(\Sigma_\psi + \lambda I)^{1/2}$, we have

$$\Sigma_\phi(\Sigma_\phi + \lambda I)^{-1} = U\Sigma_\psi(\Sigma_\psi + \lambda I)^{-1}U^T.$$

Write the SVDs $\Sigma_\phi = V_\phi D_\phi V_\phi^\top$ and $\Sigma_\psi = V_\psi D_\psi V_\psi^\top$. Then

$$D_\phi(D_\phi + \lambda I)^{-1}V_\phi^\top UV_\psi = V_\phi^\top UV_\psi D_\psi(D_\psi + \lambda I)^{-1}. \tag{4}$$

Define the diagonal matrices $\Lambda_\phi = D_\phi(D_\phi + \lambda I)^{-1}$ and $D_\psi(D_\psi + \lambda I)^{-1}$, and define the orthogonal matrix $M = V_\phi^\top UV_\psi$. Equation (4) is a homogeneous Sylvester equation:

$$\Lambda_\phi M = M\Lambda_\psi.$$

Therefore $(\Lambda_\psi)_{ii} = (\Lambda_\phi)_{jj}$ if $M_{ij} \neq 0$. Since $f : \mathbb{R}_+ \to [0, 1]$ defined by $f(x) = \frac{x}{x+\lambda}$ is invertible, this implies that $(D_\phi)_{ii} = (D_\psi)_{jj}$ if $M_{ij} \neq 0$. From this it follows that

$$(D_\phi + \lambda I)^{-1/2}M(D_\psi + \lambda I)^{1/2} = M.$$

Plugging in M and rearranging, we obtain

$$U^\top V_\phi^\top(D_\phi + \lambda I)^{-1/2}V_\phi U = V_\psi(D_\psi + \lambda I)^{-1/2}V_\psi^\top,$$

which simplifies to

$$U^\top(\Sigma_\phi + \lambda I)^{-1/2}U = (\Sigma_\psi + \lambda I)^{-1/2}.$$

By combining this with the guarantee from Claim 1 that $\phi(X) = (\Sigma_\phi + \lambda I)^{1/2}U(\Sigma_\psi + \lambda I)^{-1/2}\psi(X)$ almost surely, we obtain

$$\phi(X) = U\psi(X),$$

almost surely. This shows the converse direction of the theorem. ☐

We conclude with a proof of the claim.

Proof of Claim 1. Let $(X_1, \ldots, X_n, \ldots)$ be an infinite sequence of i.i.d copies of X. For each n, let

$$A_n = [\tilde{\phi}(X_1), \ldots, \tilde{\phi}(X_n)] \in \mathbb{R}^{k \times n}, \quad B_n = [\tilde{\psi}(X_1), \ldots, \tilde{\psi}(X_n)] \in \mathbb{R}^{l \times n}.$$

Since $d_\lambda(\phi, \psi) = 0$, by the characterization of GULP in Lemma 1 we have $\tilde{\phi}(X)^\top\tilde{\phi}(X') = \tilde{\psi}^\top(X)\tilde{\psi}(X')$ almost surely, so $A_n^\top A_n = B_n^\top B_n$ almost surely. Suppose without loss of generality that $l \leq k$. Then by Theorem 7.3.11 of [HJ12], we can construct a semi-orthogonal $U_n \in \mathbb{R}^{l \times k}$ such that $A_n = U_n B_n$ almost surely. Define the event

$$E_1 = \{A_n = U_n B_n \text{ for all } n \geq 1\}.$$

Taking a union bound over countably many n, we see that E_1 holds almost surely.

Define $W = \operatorname{span}\{\tilde{\psi}(X_i)\}_{i=1}^{\infty}$. We claim that there is a deterministic vector space $V \subseteq \mathbb{R}^l$ such that $W = V$ almost surely. Let W' be an independent copy of W. Then $W \overset{d}{=} W + W'$. For any $i \in \{0, \ldots, k\}$,

$$\mathbb{P}[\operatorname{rank}(W) \leq i] = \mathbb{P}[\operatorname{rank}(W + W') \leq i] \leq \mathbb{P}[\operatorname{rank}(W) \leq i] - \mathbb{P}[\operatorname{rank}(W) \leq i, \text{ and } W' \not\subseteq W].$$

We conclude that $\mathbb{P}[\operatorname{rank}(W) \leq i, \text{ and } W' \not\subseteq W] = 0$ for all i, so $\mathbb{P}[W' \not\subseteq W] = 0$ for the two independent copies. Therefore W is deterministic, and equals V almost surely.

Let $N = \sup\{n + 1 : \operatorname{span}\{\tilde{\psi}(X_1), \ldots, \tilde{\psi}(X_n)\} = \mathbb{R}^l\} \cup \{1\}$. Define the event that N is finite,

$$E_2 = \{N < \infty\}.$$

Since we have shown that $\operatorname{span}\{\tilde{\psi}(X_i)\}_{i=1}^{\infty} = V$ almost surely, it follows that E_2 holds almost surely.

We now prove that the semi-orthogonal random matrix $U_N \in \mathbb{R}^{k \times l}$ satisfies our conditions. Under the almost-sure events E_1 and E_2, we can write $\tilde{\psi}(X_{N+1}) = \sum_{i=1}^{N} \lambda_i \tilde{\psi}(X_i)$, and it holds that

$$\tilde{\phi}(X_{N+1}) = U_{N+1}\tilde{\psi}(X_{N+1}) = \sum_{i=1}^{N} \lambda_i U_{N+1} \tilde{\psi}(X_i) = \sum_{i=1}^{N} \lambda_i \tilde{\phi}(X_i) = \sum_{i=1}^{N} \lambda_i U_N \tilde{\psi}(X_i) = U_N \tilde{\psi}(X_{N+1}).$$

Since events E_1 and E_2 hold almost surely, and X_{N+1} is independent of N and $X_1, \ldots, X_N$,

$$\mathbb{P}[\tilde{\phi}(X) = U_N \tilde{\psi}(X)] = \mathbb{P}[\tilde{\phi}(X_{N+1}) = U_N \tilde{\psi}(X_{N+1})] = 1.$$

So we conclude that there is a deterministic semi-orthogonal matrix $U \in \mathbb{R}^{k \times l}$ such that $\tilde{\phi}(X) = U\tilde{\psi}(X)$ almost surely. Finally, recall that we have assumed that Σ_ϕ and Σ_ψ are invertible. Therefore $k = \operatorname{rank}(\Sigma_\phi) = \operatorname{rank}(\Sigma_{\tilde{\phi}}) \leq \min(\operatorname{rank}(U), \operatorname{rank}(\Sigma_{\tilde{\psi}})) = \min(\operatorname{rank}(U), \operatorname{rank}(\Sigma_\psi)) = \min(\operatorname{rank}(U), l)$. We conclude that $k = l$, and $U \in \mathbb{R}^{k \times k}$ is an orthogonal transformation. □

For $\lambda = 0$, we also characterize when the GULP distance is zero. Since GULP corresponds to the CCA distance, with slightly different normalization, this is also a characterization of when the CCA distance is zero.

Lemma 3 (Characterization for when GULP is zero, for $\lambda = 0$)**.** *If $\lambda = 0$, the two representation maps $\phi : \mathbb{R}^d \to \mathbb{R}^k$ and $\psi : \mathbb{R}^d \to \mathbb{R}^l$ have zero* GULP *distance, $d_0(\phi, \psi) = 0$, if and only if $k = l$ and there exists an invertible linear transformation $M \in \mathbb{R}^{k \times k}$ such that $\phi(X) = M\psi(X)$ a.s.*

Proof. For the "easy" direction, suppose that $k = l$ and $\phi = M\psi$ for an invertible $M \in \mathbb{R}^{k \times k}$. Then $\Sigma_\phi = M\Sigma_\psi M^\top$ and $\Sigma_{\phi\psi} = M\Sigma_\psi$. Using the characterization of GULP from Proposition 1, we obtain

$$\begin{aligned} d_0^2(\phi, \psi) &= \operatorname{tr}(\Sigma_\phi^{-1}\Sigma_\phi\Sigma_\phi^{-1}\Sigma_\phi) + \operatorname{tr}(\Sigma_\psi^{-1}\Sigma_\psi\Sigma_\psi^{-1}\Sigma_\psi) - 2\operatorname{tr}(\Sigma_\phi^{-1}\Sigma_{\phi\psi}\Sigma_\psi^{-1}\Sigma_{\phi\psi}^\top) \\ &= \operatorname{tr}(I_k) + \operatorname{tr}(I_k) - 2\operatorname{tr}((M^{-1})^\top\Sigma_\psi^{-1}M^{-1}M\Sigma_\psi\Sigma_\psi^{-1}\Sigma_\psi(M^{-1})^\top) \\ &= k + k - 2\operatorname{tr}(I_k) \\ &= 0. \end{aligned}$$

For the converse direction, we construct the representations $\tilde{\phi} = \Sigma_\phi^{-1/2}\phi$ and $\tilde{\psi} = \Sigma_\psi^{-1/2}\psi$. By the characterization of GULP in Lemma 1, the condition $d_0(\phi, \psi) = 0$ implies that $\tilde{\phi}(X)^\top\tilde{\phi}(X') = \tilde{\psi}(X)^\top\tilde{\psi}(X')$, almost surely over independent $X, X' \sim P_X$. Therefore, analogous reasoning to Claim 1 applies, and implies that $k = l$ and that there is an orthogonal transformation U such that $\tilde{\phi}(X) = U\tilde{\psi}$ almost surely. So $\phi(X) = \Sigma_\phi^{1/2}U\Sigma_\psi^{-1/2}\psi(X)$, almost surely. □

A.3 Convergence of plug-in estimator, proof of Theorem 3

In order to prove Theorem 3, we first show the following lemma.

Lemma 4. *There is a universal constant $C > 0$, such that for any B such that $\|\phi(X)\|^2, \|\psi(X)\|^2 \leq B$ almost surely, and for any $\lambda > 0$, the plug-in estimator $\hat{d}^2_{\lambda,n}$ converges to the population distance d^2_λ, with the following guarantee for any $t > 0$ and any number of samples $n > 0$,*

$$\mathbb{P}[|\hat{d}^2_{\lambda,n}(\phi,\psi) - d^2_\lambda(\phi,\psi)| \geq t + 4B^2/(n\lambda^2)] \leq \exp(-Cnt^2\lambda^4/B^4) + (k+l)\exp(-Cnt^2\lambda^6/B^6).$$

Proof. By the expanding the square and using cyclicity and linearity of the trace, similarly to the proof of Proposition 1, the plug-in estimator can alternatively be written as:

$$\hat{d}^2_{\lambda,n}(\phi,\psi) = \frac{1}{n^2}\sum_{i,j=1}^{n}(\phi(X_i)^\top(\hat{\Sigma}_\phi + \lambda I)^{-1}\phi(X_j) - \psi(X_i)^\top(\hat{\Sigma}_\psi + \lambda I)^{-1})\psi(X_j))^2. \tag{5}$$

For the analysis, also define the plug-in estimator, but with the true covariance matrices,

$$\tilde{d}^2_{\lambda,n}(\phi,\psi) = \frac{1}{n^2}\sum_{i,j=1}^{n}(\phi(X_i)^\top(\Sigma_\phi + \lambda I)^{-1}\phi(X_j) - \psi(X_i)^\top(\Sigma_\psi + \lambda I)^{-1})\psi(X_j))^2. \tag{6}$$

We bound the error between the plug-in estimator and the true distance by the triangle inequality:

$$|\hat{d}^2_{\lambda,n}(\phi,\psi) - d^2_\lambda(\phi,\psi)| \leq \underbrace{|\hat{d}^2_{\lambda,n}(\phi,\psi) - \tilde{d}^2_{\lambda,n}(\phi,\psi)|}_{\text{Term 1}} + \underbrace{|\tilde{d}^2_{\lambda,n}(\phi,\psi) - d^2_\lambda(\phi,\psi)|}_{\text{Term 2}}. \tag{7}$$

We bound Term 1 and Term 2 separately, stating our bounds in the following claims.

Claim 2 (Bound on Term 1)**.** *Under the conditions of Lemma 4, for any $t > 0$,*

$$\mathbb{P}[|\hat{d}^2_{\lambda,n}(\phi,\psi) - \tilde{d}^2_{\lambda,n}(\phi,\psi)| \geq t] \leq (k+l)e^{-nt^2\lambda^6/(2048B^6)}$$

Proof. For any $i, j \in [n]$, define $\hat{T}_{ij,\phi} = \phi(X_i)^\top(\hat{\Sigma}_\phi + \lambda I)^{-1}\phi(X_j)$ and $T_{ij,\phi} = \phi(X_i)^\top\Sigma_\phi + \lambda I)^{-1}\phi(X_j)$. We have

$$|\hat{T}_{ij,\phi} - T_{ij,\phi}| \leq B\|(\hat{\Sigma}_\phi + \lambda I)^{-1} - (\Sigma_\phi + \lambda I)^{-1}\|,$$

and

$$|\hat{T}_{ij,\phi}|, |T_{ij,\phi}| \leq B\|(\hat{\Sigma}_\phi + \lambda I)^{-1}\| \leq B/\lambda.$$

Analogous definitions and inequalities hold if we replace ϕ by ψ. Therefore,

$$\begin{aligned}
|\hat{d}^2_{\lambda,n}(\phi,\psi) - &\tilde{d}^2_{\lambda,n}(\phi,\psi)| \\
&= |\frac{1}{n^2}\sum_{i,j=1}^{n}(\hat{T}_{ij,\phi} - \hat{T}_{ij,\psi})^2 - (T_{ij,\phi} - T_{ij,\psi})^2| \\
&= |\frac{1}{n^2}\sum_{i,j=1}^{n}(\hat{T}_{ij,\phi} - \hat{T}_{ij,\psi} - T_{ij,\phi} + T_{ij,\psi})(\hat{T}_{ij,\phi} - \hat{T}_{ij,\psi} + T_{ij,\phi} - T_{ij,\psi})| \\
&\leq 4B^2(\|(\hat{\Sigma}_\phi + \lambda I)^{-1} - (\Sigma_\phi + \lambda I)^{-1}\| + \|(\hat{\Sigma}_\psi + \lambda I)^{-1} - (\Sigma_\psi + \lambda I)^{-1}\|)/\lambda.
\end{aligned}$$

So the bound on Term 1 follows from combining with the following technical claim:

Claim 3. *For any $t > 0$,*

$$\mathbb{P}[\|(\hat{\Sigma}_\phi + \lambda I)^{-1} - (\Sigma_\phi + \lambda I)^{-1}\| \geq t] \leq ke^{-nt^2\lambda^4/(32B^2)}. \tag{8}$$

$$\mathbb{P}[\|(\hat{\Sigma}_\psi + \lambda I)^{-1} - (\Sigma_\psi + \lambda I)^{-1}\| \geq t] \leq le^{-nt^2\lambda^4/(32B^2)}. \tag{9}$$

□

Proof of Claim 3. We prove the claim for ϕ, since the reasoning for ψ is analogous. First, let us prove that $\hat{\Sigma}_\phi$ concentrates around Σ_ϕ in operator norm. For each $i \in [n]$, let $Z_i = \frac{1}{n}\left(\phi(X_i)\phi(X_i)^\top - \Sigma_\phi\right)$, which is self-adjoint, satisfies $\mathbb{E}[Z_i] = 0$ and has operator norm bounded by $\|Z_i^2\| \le \frac{1}{n^2}\left(2\|\phi(X_i)\phi(X_i)^\top\|^2 + 2\|\Sigma_\phi\|^2\right) \le 4B^2/n^2$ almost surely. So applying the matrix Hoeffding inequality (Theorem 1.3 of [Tro12]) to $\hat{\Sigma}_\phi = \sum_{i=1}^n Z_i$, we have, for any $t > 0$,

$$\mathbb{P}[\|\hat{\Sigma}_\phi - \Sigma_\phi\| \ge t] \le ke^{-t^2 n/(32B^2)}.$$

Now let us show that $(\hat{\Sigma}_\phi + \lambda I)^{-1}$ concentrates to $(\Sigma_\phi + \lambda I)^{-1}$ in operator norm. Since $0 \lesssim \hat{\Sigma}_\phi, \Sigma_\phi$, for any $v \in \mathbb{R}^k$, we have

$$\begin{aligned}
\|((\hat{\Sigma}_\phi + \lambda I)^{-1} - (\Sigma_\phi + \lambda I)^{-1})v\| &\le \frac{1}{\lambda}\|(I - (\hat{\Sigma}_\phi + \lambda I)(\Sigma_\phi + \lambda I)^{-1})v\| \\
&= \frac{1}{\lambda}\|((\hat{\Sigma}_\phi - \Sigma_\phi)(\Sigma_\phi + \lambda I)^{-1})v\| \\
&\le \frac{1}{\lambda^2}\|\hat{\Sigma}_\phi - \Sigma_\phi\|\|v\|.
\end{aligned}$$

$\square$

We now bound the second term in (7).

Claim 4 (Bound on Term 2)**.** *Under the conditions of Lemma 4, for any* $t > 0$*,*

$$\mathbb{P}[|\tilde{d}^2_{\lambda,n}(\phi,\psi) - d^2_\lambda(\phi,\psi)| \ge 4B^2/(n\lambda^2) + t] \le \exp(-t^2\lambda^4 n/(8B^4)).$$

Proof. Write $\tilde{d}^2_{\lambda,n}(\phi,\psi) = \sum_{i,j=1}^n s_{ij}$, where

$$s_{ij} = \frac{1}{n^2}(\phi(X_i)^\top(\Sigma_\phi + \lambda I)^{-1}\phi(X_j) - \psi(X_i)^\top(\Sigma_\psi + \lambda I)^{-1})\psi(X_j))^2$$

is the i, j term in the sum. Since $\|(\hat{\Sigma}_\phi + \lambda I)^{-1}\|, \|(\hat{\Sigma}_\psi + \lambda I)^{-1}\| \le 1/\lambda$, and $\|\phi(X_i)\|^2, \|\psi(X_i)\|^2 \le B$, we have almost surely

$$|s_{ij}| \le \frac{4B^2}{n^2\lambda^2}.$$

Furthermore, term s_{ij} only depends on X_i and X_j. Therefore, by McDiarmid's inequality,

$$\mathbb{P}[|\tilde{d}^2_{\lambda,n}(\phi,\psi) - \mathbb{E}[\tilde{d}^2_{\lambda,n}(\phi,\psi)]| \ge t] \le \exp(-t^2\lambda^4 n/(8B^4)), \tag{10}$$

where we have used that $|\sum_{j=1}^n s_{ij}| \le 4B^2/(n\lambda^2)$ for each i. Finally, we bound the difference between $\tilde{d}^2_{\lambda,n}$ and d^2_λ in expectation over the samples. Notice that if $i \ne j$ we have $\mathbb{E}[s_{ij}] = d^2_\lambda(\phi,\psi)/n^2$. So the only terms that can add bias are the diagonal terms s_{ii}, so

$$|d^2_\lambda(\phi,\psi) - \mathbb{E}[\tilde{d}^2_{\lambda,n}(\phi,\psi)]| \le \sum_{i=1}^n |s_{ii}| \le 4B^2/(n\lambda^2) \tag{11}$$

Combining (10) and (11) proves the claim. $\square$

Combining Claims 2 and 4 with the triangle inequality (7) proves Lemma 4.

$\square$

Theorem 3 is now a simple consequence of Lemma 4.

Proof of Theorem 3. Under the conditions of Theorem 3, we have $\|\phi(X)\|^2, \|\psi(X)\|^2 \le 1$ almost surely and $\lambda \in (0,1)$. For any $t > 0$, Lemma 4 implies

$$\mathbb{P}[|\hat{d}^2_{\lambda,n}(\phi,\psi) - d^2_\lambda(\phi,\psi)| \ge t + 4/(n\lambda^2)] \le \exp(-Cnt^2\lambda^4) + (k+l)\exp(-Cnt^2\lambda^6).$$

Let $0 < \delta \leq 1$ and let $t = \frac{2}{C\lambda^3}\sqrt{\frac{\log((k+l)/\delta)}{n}}$. Then

$$\mathbb{P}[|\hat{d}^2_{\lambda,n}(\phi,\psi) - d^2_\lambda(\phi,\psi)| \geq t + 4/(n\lambda^2)] < \delta/2 + \delta/2 = \delta\,.$$

Finally, since $\lambda \in (0,1)$ we have

$$\frac{1}{\lambda^3}\sqrt{\frac{\log((k+l)/\delta)}{n}} \gtrsim t + 4/(n\lambda^2),$$

which proves the theorem. $\square$

A.4 Transfer learning distance under kernel ridge regression

Consider comparing the predictors output by kernel ridge regression with some kernel $K(x,y) = \langle\tau(x),\tau(y)\rangle$, applied to different representations. This corresponds to the case $\mathcal{F} = \{f_\beta(\cdot) : f_\beta(x) = \beta^T\tau(x)\}$ and $r(f_\beta) = ||\beta||_2^2$. Although τ may be high or even infinite dimensional, we now show that computing GULP under this $\mathcal{F}$ requires only access to $K(\cdot,\cdot)$, and not τ directly.

This is equivalent to defining new representations $\phi' = \tau\circ\phi$ and $\psi' = \tau\circ\phi$, and computing $d_\phi(\phi',\psi')$. However, τ may be high or even infinite-dimensional; traditionally in kernel ridge regression, one only wishes to compute $K(\cdot,\cdot)$ but never τ explicitly. Here, we show that $d_\lambda(\phi,\psi)$ is computable in terms of only inner products $\langle\phi(x),\phi(y)\rangle$ and $\langle\psi(x),\psi(y)\rangle$, or put differently, that $d_\lambda(\phi,\psi)$ can be written in terms of only the kernel functions associated with ϕ and ψ. By applying this result to ϕ' and ψ', this implies we only need to access $\langle\phi'(x),\phi'(y)\rangle = \langle\tau(\phi(x)),\tau(\phi(y))\rangle = K(\phi(x),\phi(y))$.

Recall that $d_\lambda(\phi,\psi)^2 = \mathrm{tr}((\Sigma_\phi+\lambda I)^{-1}\Sigma_\phi(\Sigma_\phi+\lambda I)^{-1}\Sigma_\phi)+\mathrm{tr}((\Sigma_\psi+\lambda I)^{-1}\Sigma_\psi(\Sigma_\psi+\lambda I)^{-1}\Sigma_\psi) - 2\,\mathrm{tr}((\Sigma_\phi+\lambda I)^{-1}\Sigma_{\phi\psi}(\Sigma_\psi+\lambda I)^{-1}\Sigma_{\phi\psi}^\top)$. We prove the result for the finite sample case discussed in 3, where we approximate $\Sigma_\phi = VV^T$, $\Sigma_\psi = WW^T$. Here, V consists of all the samples $\phi(x)$, with number of columns equal to the number of samples. By the kernel trick, $(\Sigma_\phi+\lambda I)^{-1}\Sigma_\phi = (VV^T+\lambda I)^{-1}VV^T = V(V^TV+\lambda I)^{-1}V^T$. Thus:

$$\begin{aligned}\mathrm{tr}((\Sigma_\phi+\lambda I)^{-1}\Sigma_\phi(\Sigma_\phi+\lambda I)^{-1}\Sigma_\phi) &= \mathrm{tr}(V(V^TV+\lambda I)^{-1}V^TV(V^TV+\lambda I)^{-1}V^T)\\ &= \mathrm{tr}((V^TV+\lambda I)^{-1}V^TV(V^TV+\lambda I)^{-1}V^TV)\end{aligned}$$

This term is expressible in terms of only $(V^TV)_{ij}$, which only depends on $\langle\phi(x_i),\phi(x_j)\rangle$ for samples x_i and x_j. Similar reasoning holds for the term $\mathrm{tr}((\Sigma_\psi+\lambda I)^{-1}\Sigma_\psi(\Sigma_\psi+\lambda I)^{-1}\Sigma_\psi)$. Finally, consider the cross-term:

$$\begin{aligned}\mathrm{tr}((\Sigma_\phi+\lambda I)^{-1}\Sigma_{\phi\psi}(\Sigma_\psi+\lambda I)^{-1}\Sigma_{\phi\psi}^\top) &= \mathrm{tr}((VV^T+\lambda I)^{-1}VW^T(WW^T+\lambda I)^{-1}WV^T)\\ &= \mathrm{tr}(V(V^TV+\lambda I)^{-1}W^TW(W^TW+\lambda I)^{-1})\\ &= \mathrm{tr}((V^TV+\lambda I)^{-1}V^TV(W^TW+\lambda I)^{-1}W^TW)\end{aligned}$$

Again, this term is expressible only in terms of V^TV and W^TW.

B Supplementary experiments

B.1 Experimental Setup

Here we briefly describe all of the network architectures used in this paper as well as the procedure for training them. All experiments were run on Nvidia Volta V100 GPUs.

Networks on MNIST For the MNIST handwritten digit database [Den12], we initialize 400 fully-connected networks with ReLU activations. Each networks accepts a flattened 28 × 28 image (784 grayscale pixels) as input and outputs at its last layer a vector of 10 probabilities for a given digit 1-10. The number of hidden layers in the networks range from 1 to 10 and the widths of all hidden layers are constant and range from 100 to 1000 in multiples of 100. Each model architecture with a fixed width and depth is randomly initialized 4 separate times with uniform Kaiming initialization [HZRS15] and zero bias. Every network is trained for 50 epochs and a batch size of 100 on all 60,000 images of the MNIST train set using the Adam optimizer [KB14] with a learning rate of 10^{-4}.

Networks on ImageNet For the ImageNet Object Localization Challenge [KSH12], we use 37 state-of-the-art models downloaded both in untrained and pretrained form from the PyTorch database of models[4]. All models can be separated into the following classes

- ResNets: regnet_x_16gf, regnet_x_1_6gf, regnet_x_32gf, regnetx_3_2_gf, regnet_x_400mf, regnet_x_800mf, regnet_x_8gf, regnet_y_16gf, regnet_y_1_6gf, regnet_y_32gf, regnet_y_3_2gf, regnet_y_400mf, regnet_y_800mf, regnet_y_8gf, resnet18, resnext50_32x4d, wide_resnet50_2
- EfficientNets: efficientnet_b0, efficientnet_b1, efficientnet_b2, efficientnet_b3, efficientnet_b4, efficientnet_b5, efficientnet_b6, efficientnet_b7
- MobileNets: mobilenet_v2, mobilenet_v3_small, mobilenet_v3_large
- ConvNeXts: convnext_base, convnext_tiny, convnext_small, convnext_large
- Miscellaneous: alexnet, googlenet, inception, mnasnet, vgg16

All models accept 3-channel RGB images of size 224 × 224 (i.e. total dimension 3 × 224 × 224). We normalize the 1,281,119 images in the train set of ImageNet to have mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$ in each RGB channel. Every models embeds the images into a latent space with dimension ranging from 400 to 4096 depending on the architecture.

Networks on CIFAR For CIFAR [KH$^+$09], we train 16 ResNet18 architectures from independent, random initializations for 50 epochs each using the FFCV library [LIE$^+$22]. They were trained with batch size 512, learning rate 0.5 on a cyclic schedule, momentum parameter 0.9, and with weight decay parameter $5e-4$.

B.2 Relationship of GULP to other distances

Embeddings of ImageNet Figure 2 of the main text compares the CKA, CCA, and GULP distances between pairs of representations of 37 ImageNet representations, estimated from 10,000 samples. In Figure 8, we extend the comparison to PWCCA and PROCRUSTES. We note that at certain λ, our distance has near-linear relationships with PROCRUSTES and CKA.

Embeddings of MNIST In Figure 9, we repeat the same experiment for MNIST embeddings with trained fully-connected networks of depths in the range from 1 to 10, and widths in $\{200, 400, 600, 800, 1000\}$.

B.3 Convergence of the plug-in estimator

In Figure 3, we estimated the distances between $\binom{15}{2} = 105$ pairs of ImageNet networks with the plug-in estimator as we increased the number of samples n. We plotted the average relative error to the 10000-sample estimate. We supplement this result with Figure 10, which shows that for $n \geq 2000$, two independent estimates of GULP have average relative error smaller than 2%. Therefore, if there is error in the plug-in estimator it is mainly due to bias, apart from roughly 2% relative error. Since the convergence in 3 indicates that the plug-in estimator is unbiased, this reinforces our claim that the plug-in estimator concentrates quickly around the true distance.

Runtime The 12 ImageNet networks for these plots were alexnet_pretrained_rep, convnext_small_pretrained_rep, efficientnet_b0_pretrained_rep, efficientnet_b3_pretrained_rep, efficientnet_b6_pretrained_rep, inception_pretrained_rep, mobilenet_v3_large_pretrained_rep, regnet_x_1_6gf_pretrained_rep, regnet_x_400mf_pretrained_rep, regnet_y_16gf_pretrained_rep, regnet_y_3_2gf_pretrained_rep, regnet_y_8gf_pretrained_rep, subsampled from the 37 models at our disposal so as to reduce the computational burden. Generating these plots took 11 minutes with an Nvidia Volta V100 GPU. The computational cost is due to the fact that distances are computed for a range of increasing number of samples n, on 66 pairs of networks and two independent trials.

B.4 GULP captures generalization performance by linear predictors

Here we supplement the experiments of Section 4.1, which show how the GULP distance captures generalization performance by linear predictors. We provide an experiment on the UTKFace dataset

---

[4] https://pytorch.org/vision/stable/models.html#classification

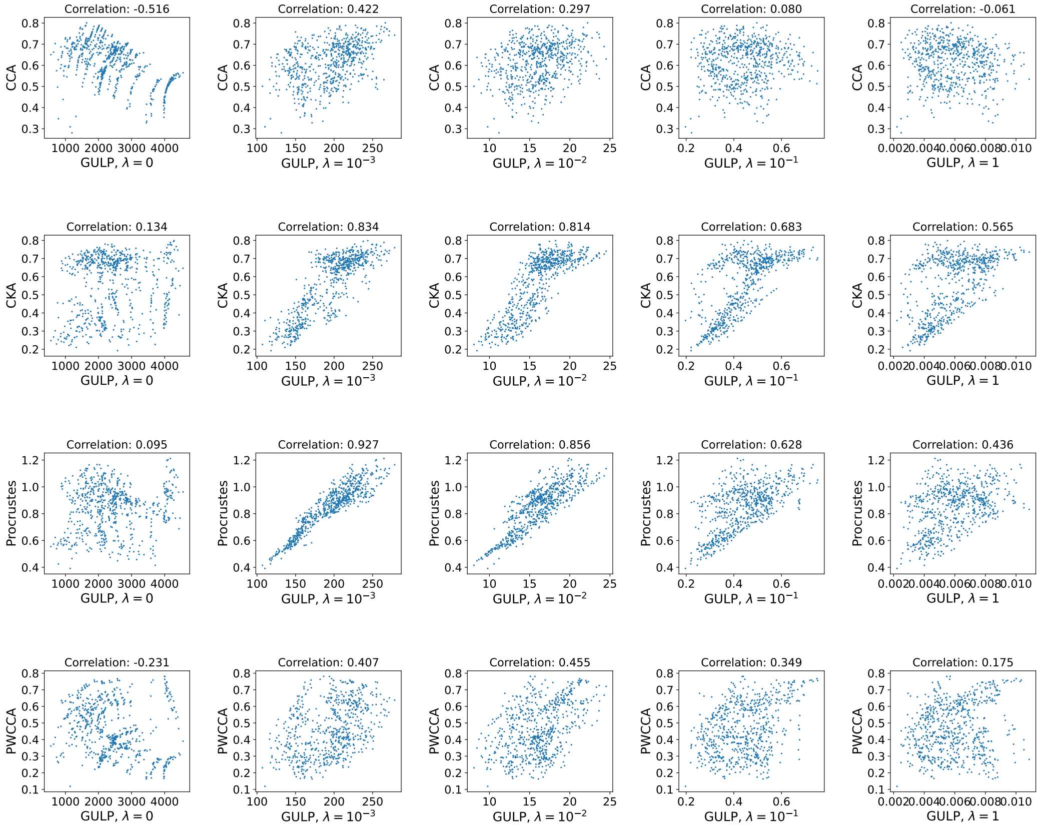

Figure 8: Scatter plots showing relationships between network distances on ImageNet. Each point is a pair of ImageNet representations, and the x and y coordinates correspond to two distances that are being compared. There is a surprising near-linear relationship between PROCRUSTES and GULP for intermediate λ. The title of each plot shows the Pearson correlation coefficient.

[ZSQ17] using the age of a face as the regression label, instead of using a random label. We consider the representation maps $\phi_1, \dots, \phi_m$ given by $m = 37$ pretrained Imagenet image classification architectures, applied to the UTKFace dataset P_X. For each pair of representations, we compute the CKA, CCA, PWCCA, and GULP distances with the plug-in estimator on 10,000 images. We then draw $n = 5000$ data points $(X_i, Y_i) \sim P_X$, where X_i is the face image and Y_i is the corresponding age. The remaining experiment details are the same as in Section 4.1. For each representation $i \in [m]$ we fit a λ-regularized least-squares linear regression to the training data $\{(X_k, Y_k)\}_{k \in [n]}$, yielding a coefficient vector $\beta_{\lambda,i}$. Finally, for each $1 \leq i \leq j \leq m$, we compute the distance τ_{ij} between predictions as an empirical average over 3000 samples in a testset. In Figure 11, we plot the Spearman ρ correlations between the prediction distances τ_{ij} and the different distances between representations (similarly to Figure 4). We run one trial, since the labels are no longer random. The GULP distance again performs favorably compared to other methods. For linear regression with $\lambda = 1$ and $\lambda = 10^{-6}$, the GULP distance with $\lambda = 1$ and $\lambda = 10^{-6}$, respectively vastly outperform previously-proposed distances in terms of predicting generalization. For linear regression with $\lambda = 10^{-4}$ and $\lambda = 10^{-2}$, GULP with $\lambda = 10^{-2}$ predicts the generalization performance on par with the CKA and PROCRUSTES distances. Notice that unlike the experiment with random labels, the best λ for GULP does not exactly match the λ used in the linear regression task, but instead is close to it.

B.5 GULP distances cluster together networks with similar architectures

Here we elaborate further on the experiments described in Section 4.2 on embeddings of MNIST networks. As described previously, we generate four independent copies of fully-connected ReLU networks with depths ranging from 1-10 and widths ranging from 100-1000. Network depth refers to

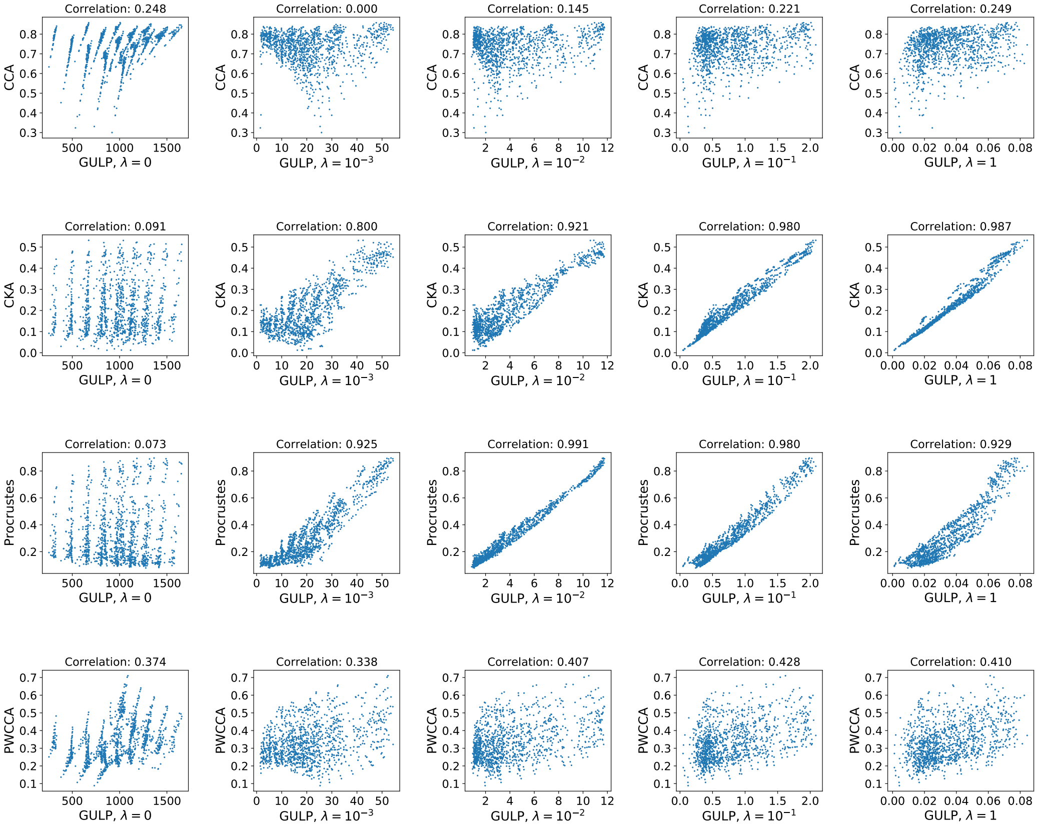

Figure 9: Scatter plots showing relationships between network distances of fully-connected network representations on MNIST. For $\lambda = 0$, there is no straight-line relationship with CCA, since the dimensions of the representations differ, and the normalization of CCA is different from that of GULP because it depends the representation dimension. Each point is a pair of MNIST representations, and the x and y coordinates correspond to two distances that are being compared. The near-linear relationship between CKA and GULP is quite evident for large λ, as it turns out that all of the kernels are closer to having the same normalization than in the case of the ImageNet dataset. Furthermore, there is a surprising near-linear relationship between CKA and PROCRUSTES for intermediate λ. The title of each plot shows the Pearson correlation coefficient.

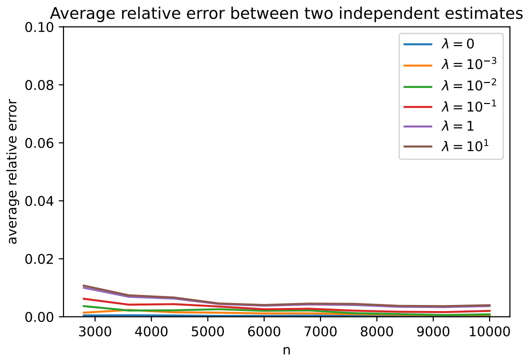

Figure 10: Relative error $|\hat{d}^{(1)}_{\lambda,n} - \hat{d}^{(2)}_{\lambda,n}|/(\hat{d}^{(1)}_{\lambda,n} + \hat{d}^{(2)}_{\lambda,n})$ between plug-in estimator on two trials $\hat{d}^{(1)}_{\lambda,n}$ and $\hat{d}^{(2)}_{\lambda,n}$ with independent samples. We have averaged across the 66 pairs of ImageNet networks. For $\lambda = 0$, due to numerical precision issues we do not plot the relative error in the estimate for $n \leq 2000$.

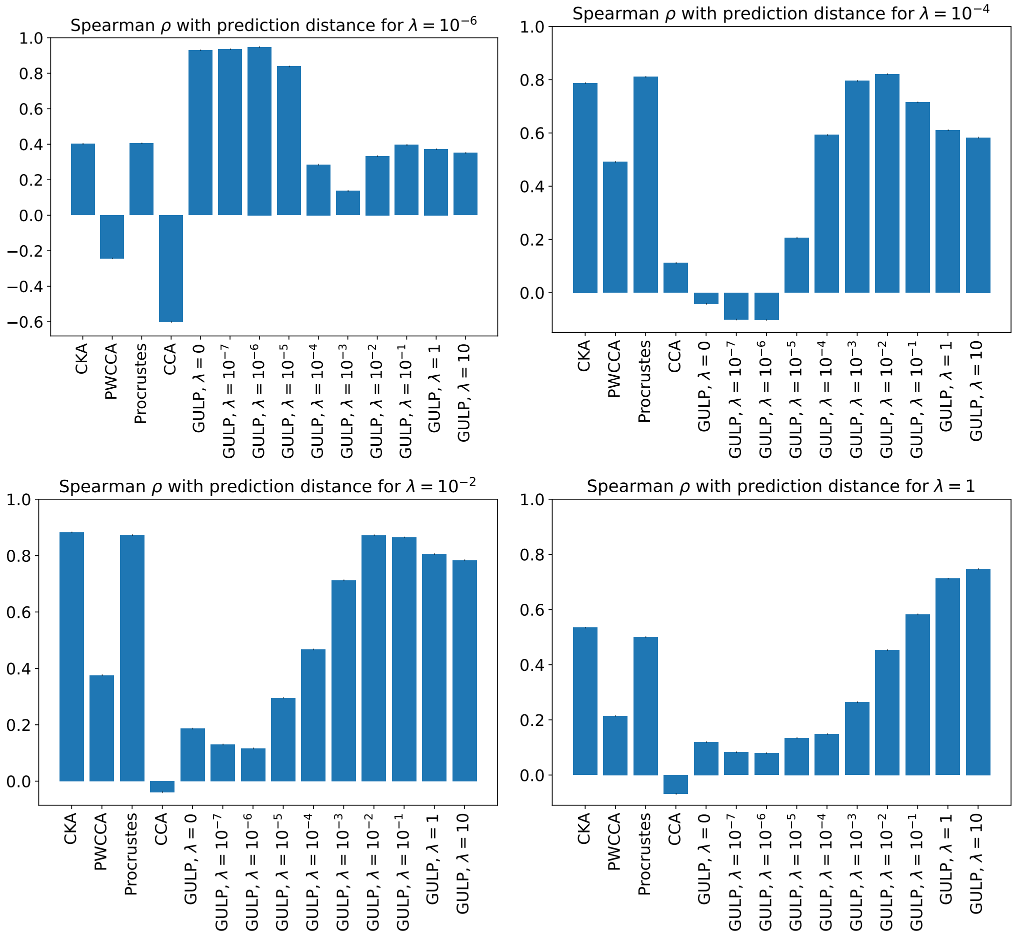

Figure 11: GULP captures generalization of linear predictors. We plot Spearman's ρ between the differences in predictions by λ-regularized linear regression, and the different distances.

the number of hidden layers in a model and network width refers to the width of each hidden layer. All networks are fully-trained on MNIST, and their last hidden layer representations are computed on 60,000 input images from the train set. For every pair of widths and depths (w_1, d_1) and (w_2, d_2), there are four trained networks with dimensions (w_1, d_1) and four trained networks with dimensions (w_2, d_2). For a given metric, we compute $4 \cdot (3 - 1) = 12$ distances between the penultimate layer representations of these networks and average them. This gives us the average distance between the penultimate layer representations of a network with dimensions (w_1, d_1) and a network with dimensions (w_2, d_2). In Figure 12 (left) we show the average PWCCA, CKA, PROCRUSTES, and GULP distances between each pair of width-depth architectures for varying λ. We also display the MDS embeddings of all $4 \times 10 \times 10$ networks colored by width and depth (center and right).

In Figure 13 we perform a very similar experiment to the one above with networks trained on CIFAR10 instead of MNIST. We generate five independent copies of fully-connected ReLU networks with depths ranging from 1-5 and widths ranging from 200-1,000. All networks are fully-trained on 10,000 images of the CIFAR10 train set and their penultimate layer representations are constructed from this set of images. Figure 13 shows the average PWCCA, CKA, PROCRUSTES, and GULP distances between each pair of width-depth architectures and show the MDS embeddings of all $5 \times 5 \times 5$ networks colored by width and depth (center and right).

Now we describe in more detail how various distance metrics cluster state-of-the-art network architectures on the ImageNet Object Localization Challenge. In Figure 15 (left) we compute the CCA, PWCCA, CKA, PROCRUSTES, and GULP distances for five groups of networks: 17 ResNets, 8 EfficientNets, 4 ConvNeXts, and 3 MobileNets. These 32 networks are fully-trained on ImageNet and are given the same 10,000 input training images to form their last hidden layer representations. As discussed in Section 4.2, all distance metrics separate ResNet architectures (blue) from the EfficientNet and ConvNeXt convolutional networks (orange and red) with GULP at $\lambda = 1$ achieving the best separation between these two clusters. For convenience, in Figure 14 we reproduce the tSNE visualizations and hierarchical clusterings of distances between pretrained ImageNet networks shown originally in 6. To further quantify the compactness of the clusterings given by these distance metrics, we compute a standard deviation ratio for each of the five network classes. Given a distance metric, this ratio is computed as the sum of squared distances between all 36 networks divided by the sum of squared distances between networks in each class:

$$\text{standard deviation ratio for class } k = \Big(\frac{1}{n(n-1)} \sum_{1\leq i\neq j\leq n} d_{ij}^2 \Big/ \frac{1}{|\mathcal{C}_k|(|\mathcal{C}_k|-1)} \sum_{i\neq j\in\mathcal{C}_k} d_{ij}^2\Big)^{\frac{1}{2}} \tag{12}$$

where $n = 36$ and $C_k \subset \{1, \ldots, n\}$ is the subset of networks in class $k = 1, \ldots, 5$. Note that a ratio of 1 implies that the size of the cluster is equal to the average distance between any two ImageNet networks. In Figure 15 (right) we plot the standard deviation ratio for each of the five network classes. As expected, the ratios under the GULP distance increase for large λ and the residual and convolutional network architectures become well separated at $\lambda = 1$. The CCA, PWCCA, CKA, and PROCRUSTES distances do not achieve the same level of separation between different network architectures but are similar to the GULP distance at $\lambda = 10^{-2}$.

Now we study distances between the same ImageNet models when they are untrained and are at random initialization. Again there are 32 untrained networks consisting of 17 ResNets, 8 EfficientNets, 4 ConvNeXts, and 3 MobileNets. Each of the untrained networks is randomly initialized ten separate times and is given the same 10,000 input training images from ImageNet. We compute the CCA, PWCCA, CKA, PROCRUSTES, and GULP distances between their penultimate layer representations which are displayed in Figure 16 (left). The distances between these networks are visualized using a two-dimensional t-SNE embedding and the standard deviation ratio (12) of each of the four groups is calculated [Figure 16 (center and right)]. Under all distance metrics we see that the ResNets (blue), EfficientNets (orange), and ConvNeXts (red) all form their own clusters. As evidenced by the standard deviation ratios, the ConvNeXt networks under the GULP distance form a tighter cluster as λ increases. Both CKA and GULP with $\lambda = 1$ achieve the most compact clusterings of ResNets, EfficientNets, and ConvNeXts.

In Figure 17 for several distance metrics we display the standard deviation ratios for the five network groups before and after training. On untrained and pretrained networks, CKA and PROCRUSTES are competitive with GULP at clustering ResNet, EfficientNet, and ConvNeXt architectures. However on ConvNeXt models, for untrained networks GULP achieves the highest standard deviation ratio with large λ and for pretrained networks it achieves the highest standard deviation ratio at intermediate values of λ.

B.6 GULP does not strongly depend on input data distribution

Here we test how the GULP distance between network architectures depends on the distribution of the input data X from which the last hidden layer representations are computed. In Figure 1 we showed a t-SNE embedding of the GULP distance ($\lambda = 10^{-2}$) between the last hidden layer representations of 37 networks pretrained on ImageNet. These penultimate layer representations were computed by passing 10,000 images from the ImageNet train set into each network. In Figure 18 we repeat this experiment and generate a t-SNE embedding of the GULP distance ($\lambda = 10^{-2}$) between ImageNet networks where each network is passed in 10,000 images from the MNIST train set. In order to input MNIST grayscale images into these networks, we convert them to RGB images where each channel has a copy of the same image and is centered and normalized as described in Section B.1. Even though all 37 networks were trained on the ImageNet train set, GULP is able to separately cluster EfficientNet, ResNet, and ConvNeXt architectures from their last hidden layer representations of MNIST images. In Figure 19 we show yet another example of this phenomenon, where GULP properly clusters ImageNet architectures when their last hidden layer representations are constructed from 10,000 face input images taken from the UTKFace train dataset [ZSQ17]. This shows that in practice the GULP distance consistently captures the same relationships between network architectures and does not strongly depend on the input data distribution used to build the network representations.

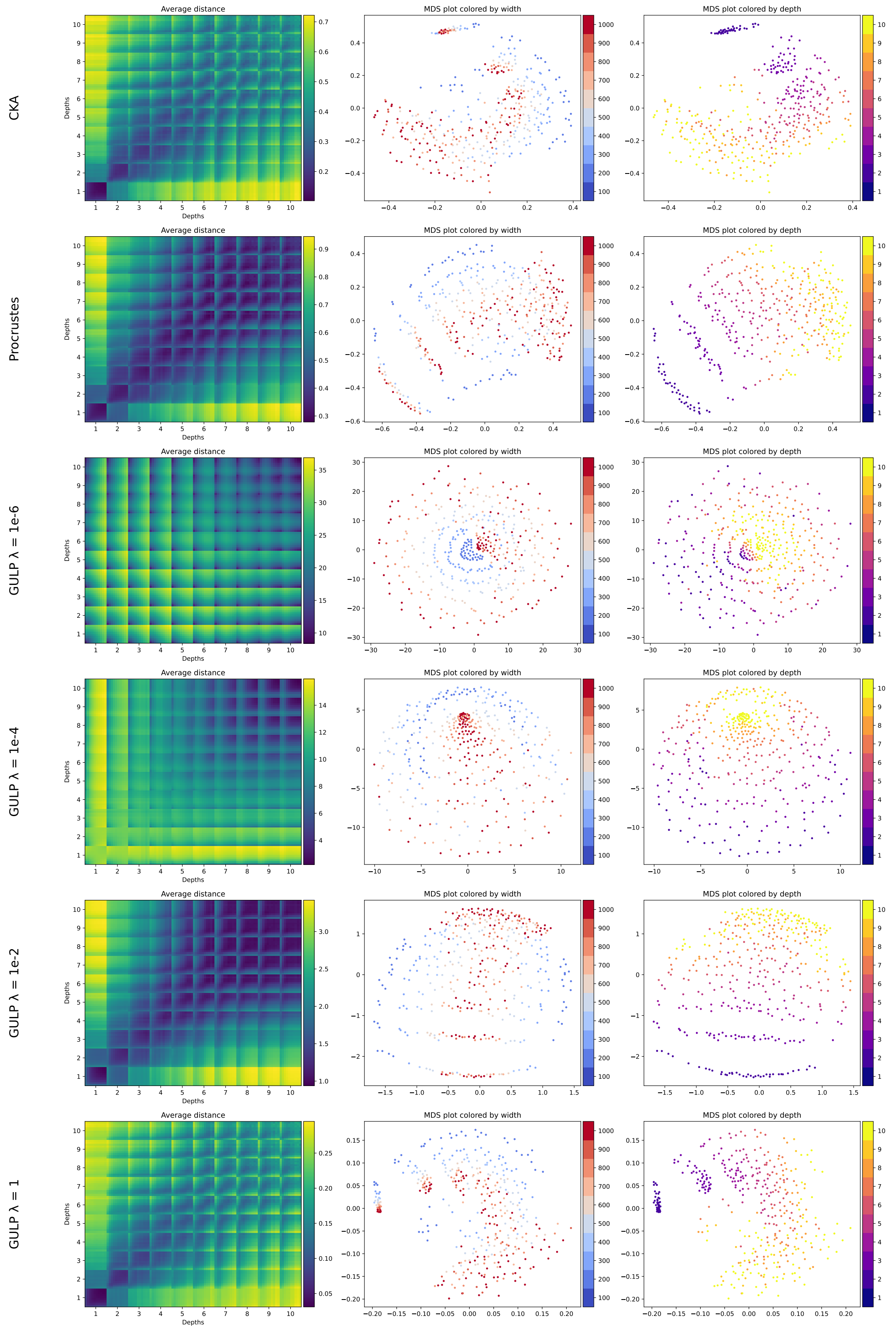

Figure 12: Average CKA, PROCRUSTES, and GULP distance between last hidden layer representations of two fully-connected ReLU networks with a given width and depth (left). Networks are fully-trained on MNIST and penultimate layer representations are constructed from 60,000 input train images. Ordering of networks along rows and columns of distance matrices has outer indices as network depths 1-10 and inner indices as network widths 100-1000. Two dimensional MDS embedding plots (center and right) of all networks colored by architecture width and depth.

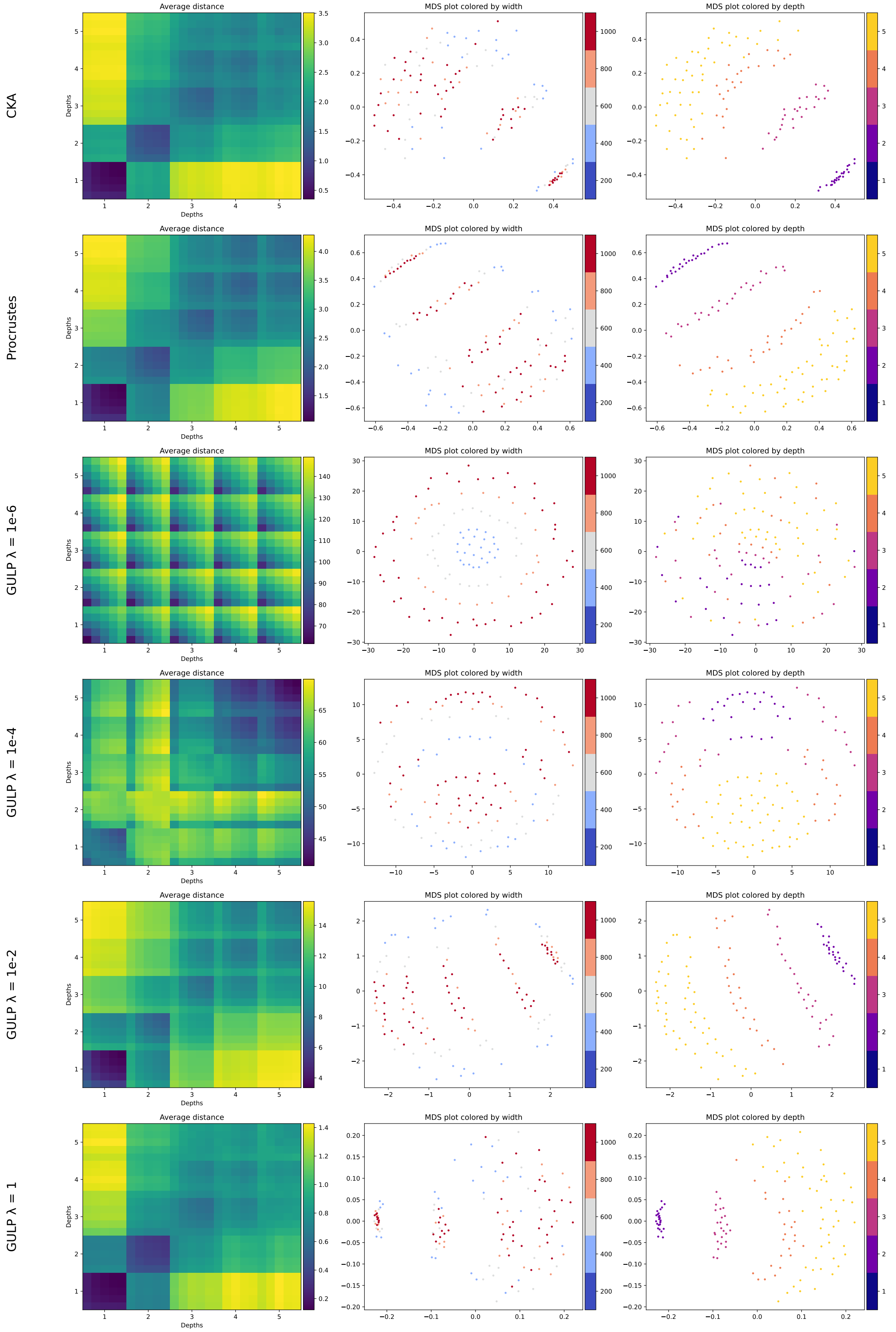

Figure 13: Average CKA, PROCRUSTES, and GULP distance between last hidden layer representations of two fully-connected ReLU networks with a given width and depth (left). Networks are fully-trained on CIFAR and penultimate layer representations are constructed from 10,000 input train images. Ordering of networks along rows and columns of distance matrices has outer indices as network depths 1-5 and inner indices as network widths 200-1000. Two dimensional MDS embedding plots (center and right) of all networks colored by architecture width and depth.

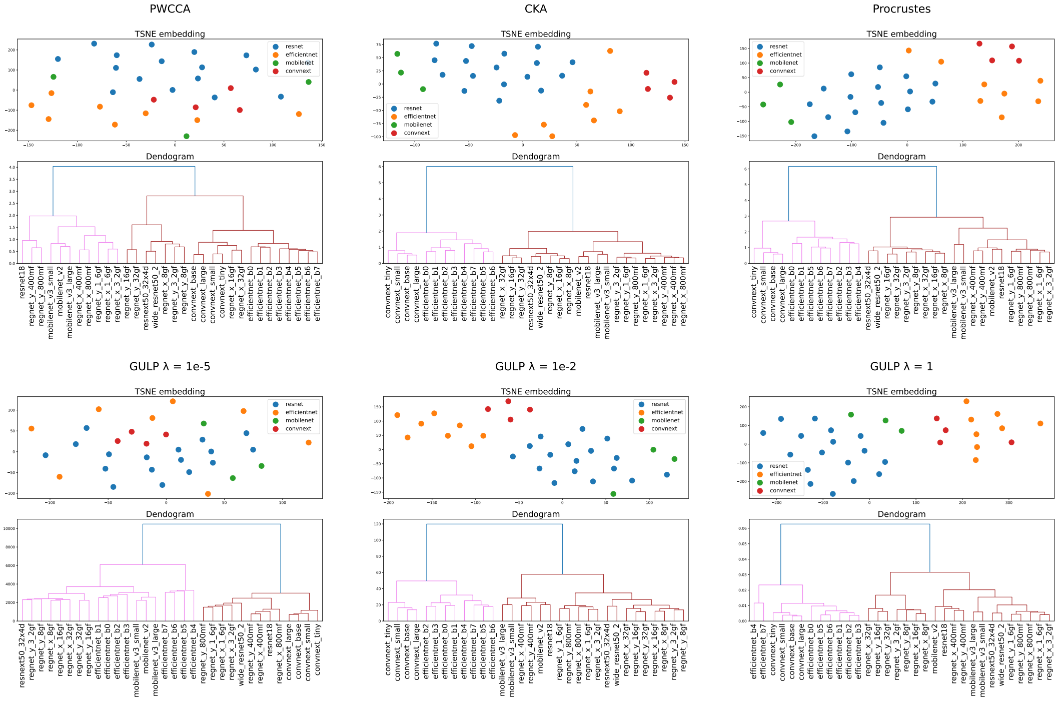

Figure 14: Reproduction of Figure 6. Embeddings of PWCCA, CKA, PROCRUSTES, and GULP distances between the penultimate layer representations of 36 pretrained ImageNet models along with their hierarchical clusterings.

B.7 Network representations converge in GULP distance during training

Here, we repeat Figure 7, but plot each distance separately and with a greater variety of regularization values λ (see Figure 20).

B.8 GULP distance at intermediate network layers

Throughout this paper, we have primarily used GULP to compare neural networks using their last hidden layer representations. Here we study how the GULP distance compares intermediate hidden layers of neural networks. Namely, we take 10 NLP BERT base models from Zhong et al. [ZGKS21] which are pretrained with different random initializations on sentences from the Multigenre Natural Language Inference (MNLI) dataset [WNB17]. Each model has 12 hidden layers and we save the representations at every hidden layer on 3,857 MNLI input train samples. In Figure 21 we plot the distance matrices for GULP at varying values of λ between every pair of hidden layers across 10 BERT networks. We also plot the tSNE, MDS, and UMAP embeddings with each colored line representing one of the 10 BERT models. In each embedding plot, earlier layers are drawn as points with a dark hue while layers closer to the end of the network are represented by points with a faded color. As expected, for each of the BERT model the GULP distances arrange their hidden layers linearly in order from their input layer to their output layer. When λ is small, the earlier layers of all 10 networks are grouped together while the later layers have large GULP distances between all 10 models. As λ increases, the later layers of all 10 models also become grouped together and GULP arranges all BERT models linearly in the order of their hidden layers. Therefore, tuning the λ parameter in GULP allows us to make distinctions between earlier and later layers of a network architecture.

B.9 Specificity versus sensitivity of GULP

Here we run three benchmark experiments of [DDS21] to compare the sensitivity and specificity of our GULP distance to CCA, PWCCA, CKA, and PROCRUSTES.

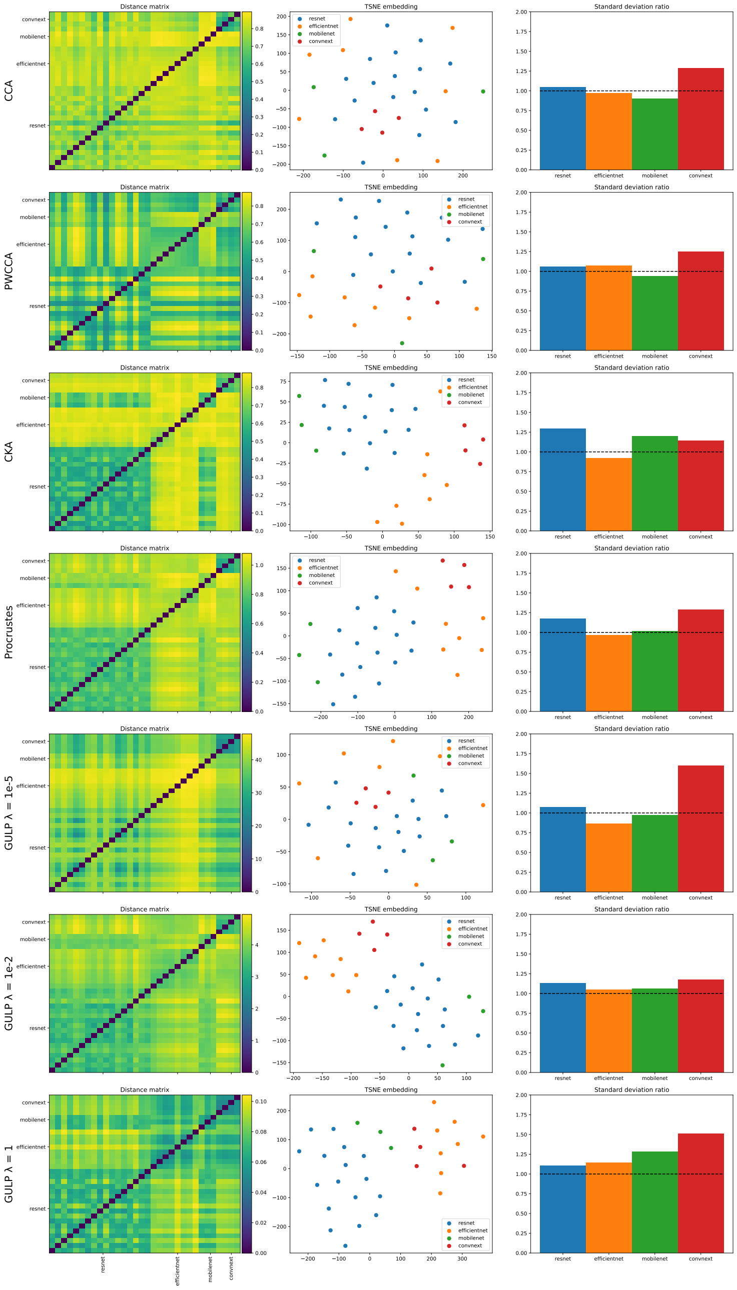

Figure 15: CCA, CKA, PROCRUSTES, and GULP distances between last hidden layer representations of 36 pretrained ImageNet networks. Representations are formed by passing 10,000 train images from ImageNet into each network. For five groups of pretrained networks (ResNet, EfficientNet, MobileNet and ConvNeXt), we compute their distance matrices (left) and two-dimensional t-SNE embeddings (center). Separation of the five network groups is quantified by their standard deviation ratios which measure the the standard deviation of the distance across all networks divided by the standard deviation of the distance in a given group. GULP, CKA, and PROCRUSTES successfully separate all four network types from each other.

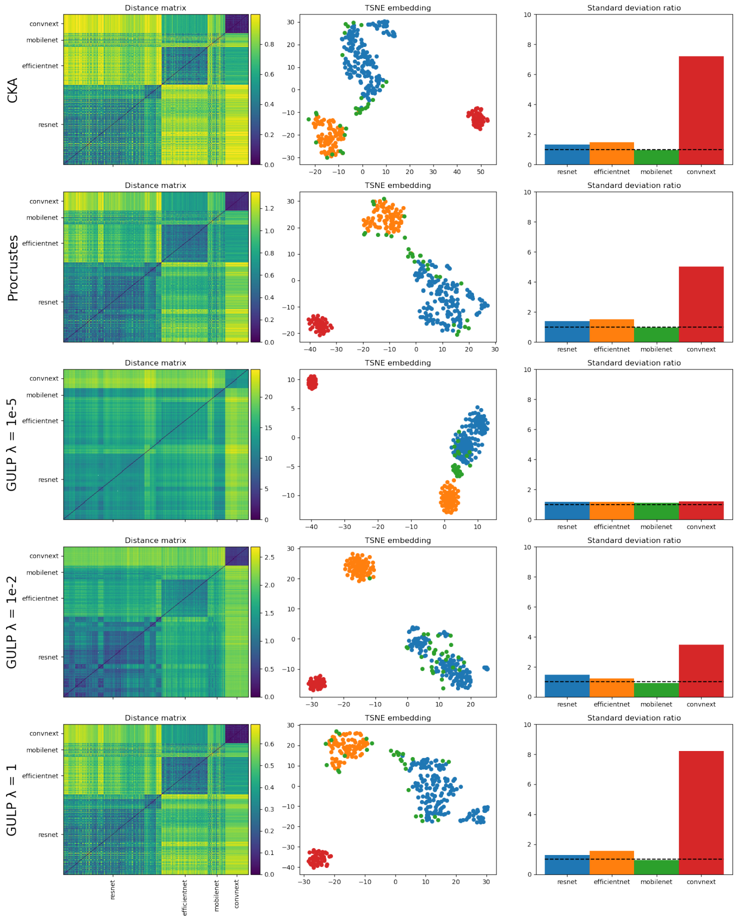

Figure 16: CKA, PROCRUSTES, and GULP distances between penultimate layer representations of 32 untrained ImageNet networks where each network model is randomly intialized 10 times. Representations are formed by passing 10,000 train images from ImageNet into each network. For four groups of pretrained networks (ResNet, EfficientNet, MobileNet, ConvNeXt), we compute their distance matrices (left) and two-dimensional t-SNE embeddings (center). Separation of the four network groups is similarly quantified by their standard deviation ratios which measure the the standard deviation of the distance across all networks divided by the standard deviation of the distance in a given group. Under all distance metrics ResNets, EfficientNets, and ConvNeXts are clustered separately with CKA and GULP at $\lambda = 1$ forming the most compact clusters.

In the first experiment, we take 10 BERT base models from Zhong et al. [ZGKS21] which are pretrained with different random initializations on sentences from the Multigenre Natural Language Inference (MNLI) dataset [WNB17]. All BERT base models have 12 hidden layers of transformer blocks with dimension 768 [DCLT18]. For each of the 10 networks, at each of the 12 layers we save the representations on 3,857 MNLI input train samples. We compute the probing accuracies of all 120 representations on the Question-answering Natural Language Inference dataset (QNLI) [WSM+18] and the Stanford Sentiment Tree Bank Task (SST-2) [SPW+13]. For a given dataset (QNLI and SST-2), we find the representation $X^* \in \mathbb{R}^{768\times 3857}$ which has the best probing accuracy and we compare the accuracies of all 120 representations to it. For every representation $X \in \mathbb{R}^{768\times 3857}$, the difference in probing accuracy from the best representation X^* is correlated with the distance between between the two represention $d(X, X^*)$ under a given distance metric (CCA, CKA, PROCRUSTES, etc.). In Figure 22 we display Spearman's ρ and Kendall's τ rank correlations of the CCA, PWCCA, PROCRUSTES, CKA, and GULP distances against the probing accuracy differences between two representations. On the QNLI dataset we see in Figure 22 (left) that GULP with large λ outperforms all other metrics including CKA and achieves the largest rank correlations with statistically significant p-values that are below 0.05. Similar results are obtained on the SST-2 dataset as seen in Figure 22

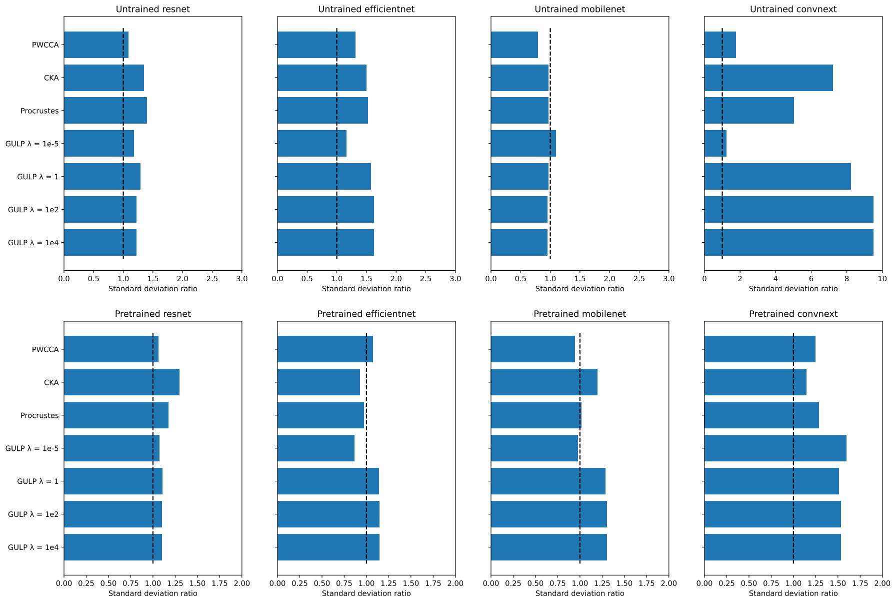

Figure 17: Standard deviation ratio of distances for five groups of architectures (ResNet, EfficientNet, MobileNet, and ConvNeXt) both for untrained and pretrained networks.

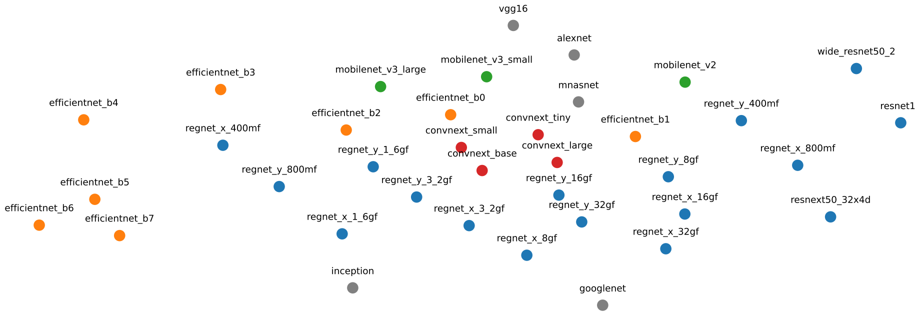

Figure 18: t-SNE embedding of penultimate layer representations of pretrained ImageNet networks with GULP distance ($\lambda = 10^{-2}$), colored by architecture type (gray denotes architectures that do not belong to a family). For each network pretrained on ImageNet we input MNIST images and compute their last hidden layer representations. Even though these ImageNet networks were not trained on MNIST data, the GULP distance is able to cluster their penultimate layer representations and consistently forms groups of MobileNet, EfficientNet, ResNet, and ConvNeXt architectures. This indicates that the GULP metric does not depend strongly on the data distribution which networks are trained on.

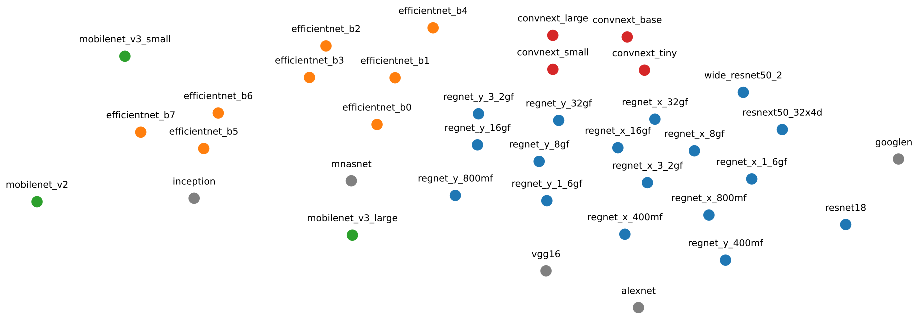

Figure 19: t-SNE embedding of penultimate layer representations of pretrained ImageNet networks with GULP distance ($\lambda = 10^{-2}$), colored by architecture type (gray denotes architectures that do not belong to a family). Contrary to Figure 1, here for each network pretrained on ImageNet we input 10,000 face images from the UTKFace train dataset and compute their last hidden layer representations. Even though these ImageNet networks were not trained on UTKFace data, the GULP distance is able to cluster their last hidden layer representations and consistently forms groups of MobileNet, EfficientNet, ResNet, and ConvNeXt architectures. This in conjunction with Figure 18 shows that the GULP metric is not overly sensitive to the input data distribution from which network representations are constructed.

(right). This shows that the GULP distance with large λ has better specificity (is less sensitive) to random initializations of a network as this has less of an effect on its correlation with probing accuracy compared to the other metrics.

In the second experiment, we study 50 BERT base models from McCoy et al. [MML19] which are trained on MNLI and finetuned for classification with different finetuning seeds at initialization. Similar to the experiment above, we compute 600 representations of the 50 BERT models at each of the 12 layers using 3,857 MNLI input train samples. We are interested in studying how distances between these representations correlate with their out-of-distribution (OOD) performance on a different task. Namely, as our measure of OOD performance we compute each representation's accuracy on the "Lexical Heuristic (Non-entailment)" subset of the HANS dataset [MPL19]. As before, we choose the best representation X^* with the lowest OOD accuracy. Then for every representation X the difference in OOD accuracy from the best representation X^* is correlated with the distance between between the two represenions $d(X, X^*)$ under a given distance metric. Spearman's ρ and Kendall's τ rank correlations of the CCA, PWCCA, PROCRUSTES, CKA, and GULP distances are shown in Figure 23. Note that CCA, PWCCA, PROCRUSTES, and GULP with small λ have the largest correlation with OOD accuracy. Since the BERT model representations were constructed on in-distribution MNLI data, this implies that these distance metrics can detect differences between OOD accuracy of different models without access to OOD data.

Lastly, for the third experiment we study 100 BERT medium models taken from Zhong et al. [ZGKS21] which are fully-trained on the MNLI dataset with 10 pretraining seeds and further finetuned on MNLI with 10 different finetuning seeds by Ding et al. [DDS21]. Each BERT medium model has 8 hidden layers of width 512 [DCLT18]. We study the OOD accuracy of these models on the antonymy stress test and the numerical stress test defined in Naik et al. [NRS+18]. As with the previous experiments, we compute 800 representations of the 100 BERT models at each of the 8 layers using 3,857 MNLI input train samples. For every representation X the difference in OOD accuracy from the best representation X^* is correlated with the distance between between the two represenions $d(X, X^*)$ under a given distance metric. Spearman's ρ and Kendall's τ rank correlations of the CCA, PWCCA, PROCRUSTES, CKA, and GULP distances are shown in Figure 24. As shown in the original experiments by Ding et al. [DDS21], none of the distance metrics show a large rank correlation with the OOD accuracy for either of the stress tests and the associated p-values are not significant at the 0.05 level except for GULP with $\lambda > 10^{-2}$.

In summary, these benchmark experiments show that the GULP distance exhibits specificity (is not sensitive) to random initializations of a network as shown in Figure 22 and this become particularly

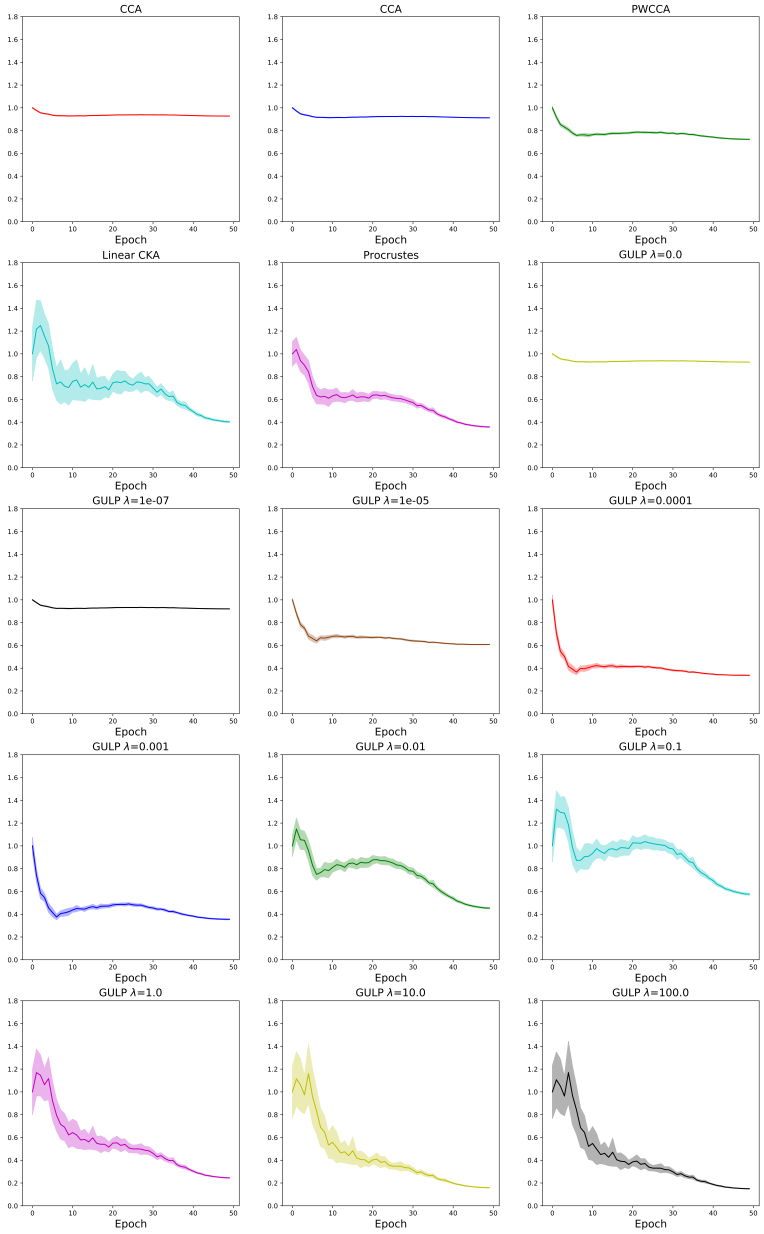

Figure 20: The empirical distances between penultimate layer representations of 16 independently trained ResNet18 architectures during training, computed using $3,000$ samples and averaged over all pairs. Distances are scaled by their average value at iteration 0 for the sake of comparison between metrics.

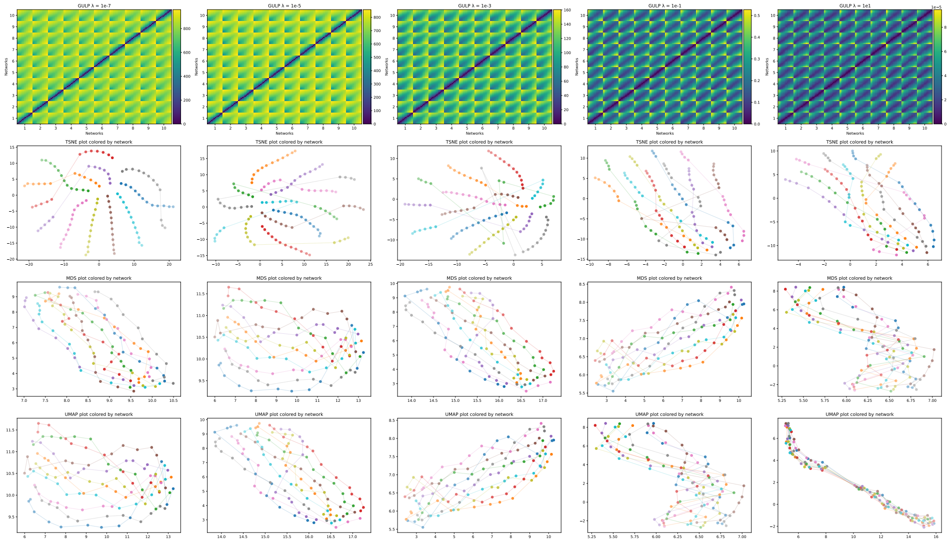

Figure 21: Top row shows GULP distance matrices between 12 hidden layers of 10 fully-trained NLP BERT base models with different random initializations. Representations at every hidden layer are constructed from 3,857 MNLI input train samples which are then used to compute the GULP distance between every pair of layers across the 10 models. Distance matrices are embedded using tSNE, MDS, and UMAP where each colored line represents one of the 10 BERT models. Earlier layers are drawn as dark saturated points while layers close to the output of the network are drawn as faded points. For each of the 10 BERT networks, GULP finds a one-dimensional embedding of its layers which respects their ordering. Across all BERT models, GULP with small λ groups together the earlier layers of the 10 network architectures but assigns large distances between the later layers. This is particularly emphasized in the top left tSNE embedding. As λ increases, the later layers of all 10 models also become grouped together until all BERT networks are linearly aligned in the order of their hidden layers.

apparent at large λ. Additionally, it is sensitive to the out-of-distribution accuracy of a model as supported by Figure 23 where it improves upon the performance of CCA, PWCCA, and PROCRUSTES.

B.10 GULP distances do *not* especially capture generalization on logistic regression

In this section, we provide Figure 25, which replicates the experiment of Figure 4, but where the downstream transfer learning task is binary logistic regression instead of ridge regression. We assign labels of 0 and 1 with equal probability, and compute the resultant test prediction accuracy averaged over 3000 samples. We find (perhaps unsurprisingly) that GULP, as defined for ridge regression, does not capture downstream generalization better than baselines on logistic regression tasks. This motivates the extension of GULP to logistic regression in future work.

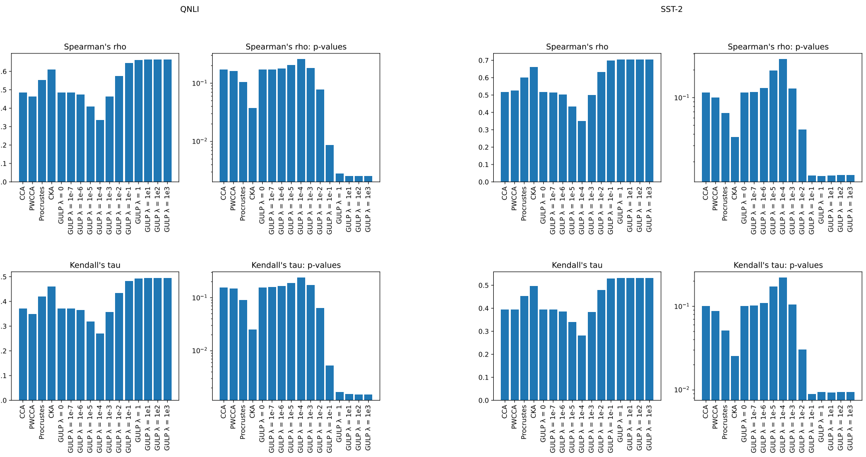

Figure 22: Spearman's ρ and Kendall's τ rank correlations and associated p-values for difference of probing accuracy between two representations vs. distance between two representations. Representations are constructed from 12 layers of 10 BERT base models using 3,857 MNLI input train samples. Rank correlations are computed with probing accuracy on the QNLI and SST-2 datasets (left and right).

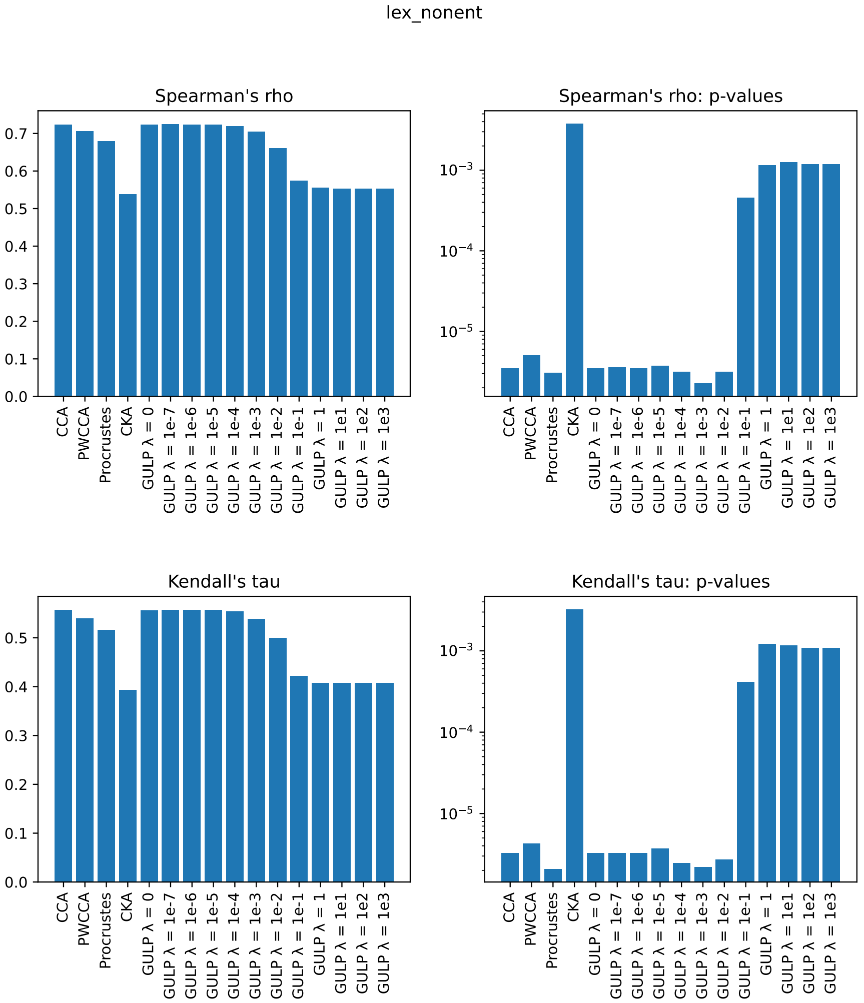

Figure 23: Spearman's ρ and Kendall's τ rank correlations and associated p-values for difference of OOD accuracy between two representations vs. distance between two representations. Representations are constructed from 12 layers of 50 BERT base models using 3,857 MNLI input train samples. The BERT base models are finetuned for classification and the OOD accuracy is computed on the "Lexical Heuristic (Non-entailment)" subset of the HANS dataset.

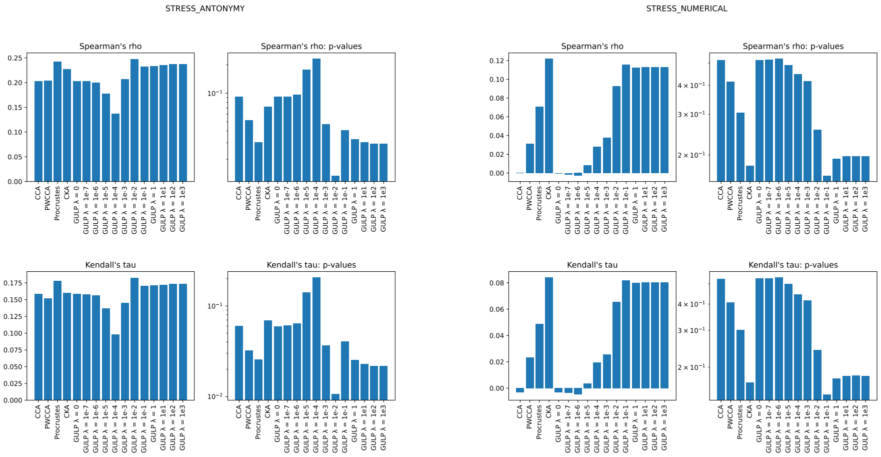

Figure 24: Spearman's ρ and Kendall's τ rank correlations and associated p-values for difference of OOD accuracy between two representations vs. distance between two representations. Representations are constructed from 8 layers of 100 BERT medium models using 3,857 MNLI input train samples. The BERT base models are trained from a combination of 10 pretraining and 10 finetuning seeds and the OOD accuracy of each model is measured on the antonymy stress and the numerical stress tests.

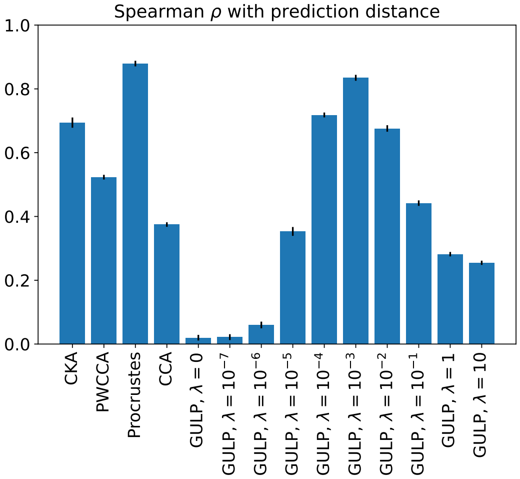

Figure 25: GULP does not capture generalization of the predictors output by logistic regression. We plot Spearman's ρ between the differences in predictions by λ-regularized linear regression, and the different distances. Results are averaged over 10 trials.