DATA-EFFICIENT MUTUAL INFORMATION NEURAL ESTIMATOR FOR STATISTICAL DEPENDENCY TESTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Measuring Mutual Information (MI) between high-dimensional, continuous, random variables from observed samples has wide theoretical and practical applications. Recent works have developed accurate MI estimators through provably low-bias approximations and tight variational lower bounds assuming abundant supply of samples, but require an unrealistic number of samples to guarantee statistical significance of the estimation. In this work, we focus on improving data efficiency and propose a Data-Efficient MINE Estimator (DEMINE) that can provide a tight lower confident interval of MI under limited data, through adding cross-validation to the MINE lower bound (Belghazi et al., 2018). Hyperparameter search is employed and a novel meta-learning approach with task augmentation is developed to increase robustness to hyperparameters, reduce overfitting and improve accuracy. With improved data-efficiency, our DEMINE estimator enables statistical testing of dependency at practical dataset sizes. We demonstrate the effectiveness of DEMINE on synthetic benchmarks and real world fMRI data, with application of inter-subject correlation analysis.

1 INTRODUCTION

Mutual Information (MI) is an important, theoretically grounded measure of similarity between random variables. MI captures general, non-linear, statistical dependencies between random variables. MI estimators that estimate MI from samples are important tools widely used in not only subjects such as physics and neuroscience, but also machine learning ranging from feature selection and representation learning to explaining decisions and analyzing generalization of neural networks.

Existing studies on MI estimation between general random variables focus on deriving asymptotic lower bounds and approximations to MI under infinite data, and techniques for reducing estimator bias such as bias correction, improved signal modeling with neural networks and tighter lower bounds. Widely used approaches include the k-NN-based KSG estimator (Kraskov et al., 2004) and the variational lower-bound-based MINE estimator family (Belghazi et al., 2018; Poole et al., 2018).

Despite the empirical and asymptotic bias improvements, MI estimation has not seen wide adoption. The challenges are two-fold. First, the analysis of dependencies among variables - let alone any MI analyses for scientific studies - requires not only an MI estimate, but also confidence intervals (Holmes & Nemenman, 2019) around the estimate to quantify uncertainty and statistical significance. Existing MI estimators, however, do not provide confidence intervals. As low probability events may still carry a significant amount of information, the MI estimates could vary greatly given additional observations (Poole et al., 2018). Towards providing upper and lower bounds of true MI under limited number of observations, existing MI lower bound techniques assume infinite data and would need further relaxations when a limited number of observations are provided. Closest to our work, Belghazi et al. (2018) studied the lower bound of the MINE estimator under limited data, but it involves bounds on generalization error of the signal model and would not yield useful confidence intervals for realistic datasets. Second, practical MI estimators should be insensitive to the choice of hyperparameters. An estimator should return a single MI estimate with its confidence interval irrespective of the type of the data and the number of observations. For learning-based approaches, this means that the model design and optimization hyperparameters need to not only be determined automatically but also taken into account when computing the confidence interval.

Towards addressing these challenges, our estimator, DEMINE, introduces a predictive MI lower bound for limited samples that enables statistical dependency testing under practical dataset sizes.
Our estimator builds on top of the *MINE* estimator family, but performs cross-validation to remove the need to bound generalization error. This yields a much tighter lower bound agnostic to hyperparameter search. We automatically selected hyperparameters through hyperparameter search, and a new cross-validation meta-learning approach is developed, based upon few-shot meta-learning, to automatically decide initialization of model parameters. Meta-overfitting is strongly controlled through task augmentation, a new task generation approach for meta-learning. With these improvements, we show that DEMINE enables practical statistical testing of dependency for not only synthetic datasets but also for real world functional Magnetic Resonance Imaging (fMRI) data analysis capturing nonlinear and higher-order brain-to-brain coupling.

Our contributions are summarized as follows: 1) A data-efficient Mutual Information Neural Estimator (DEMINE) for statistical dependency testing; 2) A new formulation of meta-learning using Task Augmentation (Meta-DEMINE); 3) Application to real life, data-scarce applications (fMRI).

2 RELATED WORK

2.1 MI Estimation

A widely used approach for estimating MI from samples is using k-NN estimates, notably the KSG estimator (Kraskov et al., 2004). Gao et al. (2017) provided a comprehensive review and studied the consistency and of asymptotic confidence bound of the KSG estimator (Gao et al., 2018). MI estimation can also be achieved by estimating individual entropy terms through kernel density estimation (Ahmad & Lin, 1976) or cross-entropy (McAllester & Statos, 2018). Despite their good performance on random variables with few dimensions, MI estimation on high-dimensional random variables remains challenging for commonly used Gaussian kernels. Fundamentally, estimating MI requires accurately modeling the random variables, where high-capacity neural networks have shown excellent performance on complex high-dimensional signals such as text, image and audio.

Recent works on MI estimation have focused on developing tight asymptotic variational MI lower bounds where neural networks are used for signal modeling. The IM algorithm (Agakov, 2004) introduces a variational MI lower bound, where a neural network $q(z|x)$ is learned as a variational approximation to the conditional distribution $P(Z|X)$. The IM algorithm requires the entropy, $H(Z)$, and $E_{XZ} \log q(z|x)$ to be tractable, which applies to latent codes of Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) as well as categorical variables. Belghazi et al. (2018) introduces MI lower bounds MINE and MINE-f which allow the modeling of general random variables and shows improved accuracy for high-dimensional random variables, with application to improving generative models. Poole et al. (2018) introduces a spectrum of energy-based MI estimators based on MINE and MINE-f lower bounds and a new TCPC estimator for the case when multiple samples from $P(Z|X)$ can be drawn.

Our work introduces cross-validation to the MINE-f estimator. We derive the lower bound of MINE-f under limited number of samples, and introduce meta-learning and hyperparameter search to enable practical statistical dependency testing.

2.2 Meta Learning

Meta-learning, or “learning to learn”, seeks to improve the generalization capability of neural networks by searching for better hyperparameters (Maclaurin et al., 2015), network architectures (Pham et al., 2018), initialization (Finn et al., 2017a, 2018), Kim et al. (2018) and distance metrics (Vinyals et al., 2016). Meta-learning approaches have shown significant performance improvements in applications such as automatic neural architecture search (Pham et al., 2018), few-shot image recognition (Finn et al., 2017a) and imitation learning (Finn et al., 2017b).

In particular, our estimator benefits from the Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017a) framework which is designed to improve few-shot learning performance. A network initialization is learned to maximize its performance when fine-tuned on few-shot learning tasks. Applications include few-shot image classification and navigation.

We leverage the model-agnostic nature of MAML for MI estimation between generic random variables and adopt MAML for maximizing MI lower bounds. To construct a collection of diverse tasks for MAML learning from limited samples, inspired by MI’s invariance to invertible transformations,
we propose a task-augmentation protocol to automatically construct tasks by sampling random transformations to transform the samples. Results show reduced overfitting and improved generalization.

3 BACKGROUND

In this section, we will provide the background necessary to understand our approach. We define $X$ and $Z$ to be two random variables, $P(X, Z)$ is the joint distribution, and $P(X)$ and $P(Z)$ are the marginal distributions over $X$ and $Z$ respectively. Our goal is to estimate MI, $I(X; Z)$ given independent and identically distributed (i.i.d.) sample pairs $(x_i, z_i)$, $i = 1, 2, \ldots n$ from $P(X, Z)$. Let $\mathcal{F} = \{T_\theta(x, z)\}_{\theta \in \Theta}$ be a class of scalar functions, where $\theta$ is the set of model parameters. Let $q(x|z) = p(x) \frac{e^{T_\theta(x,z)}}{\sum_{z'} e^{T_\theta(x,z')}}$. Results from previous works (Belghazi et al., 2018; Poole et al., 2018) show that the following energy-based family of lower bounds of MI hold for any $\theta$:

$$
I(X; Z) \geq \mathbb{E}_{(x,z) \sim P_{X,Z}} \log \frac{q(x|z)}{p(x)} = \mathbb{E}_{(x,z) \sim P_{X,Z}} T_\theta(x, z) - \mathbb{E}_{x \sim P_X} \log \mathbb{E}_{z \sim P_Z} e^{T_\theta(x,z)} \triangleq I^{\text{EB1}} \\
\geq \mathbb{E}_{(x,z) \sim P_{X,Z}} T_\theta(x, z) - \mathbb{E}_{x \sim P_X} \mathbb{E}_{z \sim P_Z} e^{T_\theta(x,z)} \triangleq I^{\text{MINE}} \\
\geq \mathbb{E}_{(x,z) \sim P_{X,Z}} T_\theta(x, z) - \mathbb{E}_{x \sim P_X} \mathbb{E}_{z \sim P_Z} e^{T_\theta(x,z)} + 1 \triangleq I^{\text{MINE},t} + I^{\text{EB}}
$$

where, $\mathbb{E}$ is the expectation over the given distribution. Based on $I^{\text{MINE}}$, the MINE estimator $\hat{I}(X, Z)_n$ is defined as in Eq.2 Estimators for $I^{\text{EB1}}$, $I^{\text{MINE},t}$ and $I^{\text{EB}}$ can be defined similarly.

$$
\hat{I}(X; Z)_n = \sup_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n T_\theta(x_i, z_i) - \log \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n e^{T_\theta(x_i, z_j)}. 
$$

With infinite samples to approximate expectation, Eq.2 converges to the lower bound $\hat{I}(X, Z)_\infty = \sup_{\theta \in \Theta} \hat{I}^{\text{MINE}}$. Note that the number of samples $n$ needs to be substantially more than the number of model parameters $d = |\theta|$ to guarantee that $T_\theta(X, Y)$ does not overfit to the samples $(x_i, z_i)$, $i = 1, 2, \ldots n$ and overestimate MI. Formally, the sample complexity of MINE is defined as the minimum number of samples $n$ in order to achieve $\hat{I}(X, Z)_n$ converges to $\hat{I}(X, Z)_\infty$.

$$
\Pr(\hat{I}(X, Z)_n - \hat{I}(X, Z)_\infty \leq \epsilon) \geq 1 - \delta.
$$

Specifically, MINE proves that under the following assumptions: 1) $T_\theta(X, Z)$ is L-Lipschitz; 2) $T_\theta(X, Z) \in [-M, M]$, 3) $\theta_i \in [-K, K]$, $\forall i \in 1, \ldots, d$, the sample complexity of MINE is given by Eq.4

$$
n \geq \frac{2M^2(d \log(16KL\sqrt{\epsilon}/\delta) + 2dM + \log(2/\delta))}{\epsilon^2}.
$$

For example, a neural network with dimension $d = 10,000$, $M = 1$, $K = 0.1$ and $L = 1$, achieving a confidence interval of $\epsilon = 0.1$ with 95% confidence ($\delta = 0.05$) would require $n \geq 18,756,256$ samples. This is achievable for synthetic example generated by GANs like that studied in Belghazi et al. (2018). For real data, however, the cost of data acquisition for reaching statistically significant estimation can be prohibitively expensive. Our approach instead uses the MI lower bounds specified in Eq.1 from a prediction perspective, inspired by cross-validation. Our estimator, DEMINE, improves sample complexity by disentangling data for lower bound estimation from data for learning a generalizable $T_\theta(X, Z)$. DEMINE enables high-confidence MI estimation on small datasets.

4 APPROACH

Section 4.1 specifies DEMINE for predictive MI estimation and derives the confidence interval; Section 4.2 formulates Meta-DEMINE, explains task augmentation, and defines the optimization algorithms.

4.1 PREDICTIVE MUTUAL INFORMATION ESTIMATION

In DEMINE, we interpret the estimation of $I^{\text{MINE}}$ lower bound as a learning problem. The goal is given a limited number of samples, infer the optimal network $T_\theta^{*}(X, Z)$ with parameters $\theta^{*}$

$^1$We follow the same notations in Belghazi et al. (2018). We encourage the review of Belghazi et al. (2018); Poole et al. (2018) for a detailed understanding of $I^{\text{MINE}}$, $I^{\text{EB1}}$, and $I^{\text{EB}}$.

$^2$MINE lower bound can also be interpreted in the predictive way, but will result in a higher sample complexity than $I^{\text{MINE}}$ lower bound. We choose $I^{\text{MINE}}$ in favor of a lower sample complexity over bound tightness.

---

3
We now derive the sample complexity of DEMINE defined as the number of samples required for learning a network \( \hat{\theta} \) as an approximation to \( \theta^* \) whereas the validation set is used for computing the DEMINE estimation. DEMINE learns \( \hat{\theta} \) by maximizing the MI lower bound on the training set as follows:

\[
\hat{\theta} = \arg \min_{\theta \in \Theta} \mathcal{L}(\{(x, z)\}_{\text{train}}, \theta),
\]

where

\[
\mathcal{L}(\{(x, z)\}_{\mathcal{B}}, \theta) = -\frac{1}{|\mathcal{B}|} \sum_{i=1}^{\mathcal{|\mathcal{B}|}} T_{\theta}(x_i, z_i)_{\mathcal{B}} + \frac{1}{|\mathcal{B}|^2} \sum_{i=1}^{\mathcal{|\mathcal{B}|}} \sum_{j=1}^{\mathcal{|\mathcal{B}|}} e^{T_{\theta}(x_i, z_i)_{\mathcal{B}}} - 1.
\]

The DEMINE algorithm is shown in Algorithm 1.

**Algorithm 1 DEMINE**

**Input Data:** \( \{(x, z)_{\text{train}}, (x, z)_{\text{val}}\} \)

**Parameters:** Batch \( B \), Iterations \( N_O \), Learning rate \( \eta \)

**Output:** MI, \( T_{\theta}(X, Z) \)

1: \( \theta^{(0)} \leftarrow \) Xavier Initialization (Glorot & Bengio 2010)
2: for \( i = 1 : N_O \) do
3: Sample a batch of \( (x_i, z_i)_{\mathcal{B}} \sim (x, z)_{\text{train}} \)
4: Compute \( \mathcal{L}(\{(x_i, z_i)_{\mathcal{B}}, \theta^{(i-1)}\}) \)
5: Compute \( \nabla_{\theta} \mathcal{L} \) - gradient for \( \theta \)
6: Update \( \theta^{(i)} \) using Adam (Kingma & Ba 2014) with \( \eta \)
7: end for
8: MI = \( I(X, Z)_{\theta^{(N_O)}} \)
9: return MI, \( \theta^{(N_O)} \)

defined as follows:

\[
\theta^* = \arg \max_{\theta \in \Theta} \mathbb{E}_{P_X Z} T_{\theta}(X, Z) - \mathbb{E}_{P_X} \mathbb{E}_{P_Z} e^{T_{\theta}(X, Z)} + 1.
\]

Specifically, samples from \( P(X, Z) \) are subdivided into a training set \( \{(x_i, z_i)_{\text{train}}, i = 1, \ldots, \mathcal{m}\} \) and a validation set \( \{(x_i, z_i)_{\text{val}}, i = 1, \ldots, \mathcal{n}\} \). The training set is used for learning a network \( \hat{\theta} \) as an approximation to \( \theta^* \) whereas the validation set is used for computing the DEMINE estimation \( I(X, Z)_{\hat{\theta}} \) defined as in Eq.5

\[
\hat{I}(X, Z)_{\hat{\theta}} = \frac{1}{\mathcal{n}} \sum_{i=1}^{\mathcal{n}} T_{\theta}(x_i, z_i)_{\text{val}} - \frac{1}{\mathcal{n}^2} \sum_{i=1}^{\mathcal{n}} \sum_{j=1}^{\mathcal{n}} e^{T_{\theta}(x_i, z_i)_{\text{val}}} + 1
\]

(5)

We propose an approach to learn \( \hat{\theta} \), DEMINE. DEMINE learns \( \hat{\theta} \) by maximizing the MI lower bound on the training set as follows:

\[
\hat{\theta} = \arg \min_{\theta \in \Theta} \mathcal{L}(\{(x, z)\}_{\text{train}}, \theta),
\]

where

\[
\mathcal{L}(\{(x, z)\}_{\mathcal{B}}, \theta) = -\frac{1}{|\mathcal{B}|} \sum_{i=1}^{\mathcal{|\mathcal{B}|}} T_{\theta}(x_i, z_i)_{\mathcal{B}} + \frac{1}{|\mathcal{B}|^2} \sum_{i=1}^{\mathcal{|\mathcal{B}|}} \sum_{j=1}^{\mathcal{|\mathcal{B}|}} e^{T_{\theta}(x_i, z_i)_{\mathcal{B}}} - 1.
\]

(6)

The DEMINE algorithm is shown in Algorithm 1.

**Sample complexity analysis.** Because \( \hat{\theta} \) is learned independently of validation samples \( \{(x_i, z_i)_{\text{val}}, i = 1, \ldots, \mathcal{n}\} \), the sample complexity of the DEMINE estimator does not involve the model class \( \mathcal{F} \) and the sample complexity is greatly reduced compared to MINE-f. DEMINE estimates \( I(X, Z)_{\infty, \hat{\theta}} \) when infinite number of samples are provided, defined as:

\[
\hat{I}(X, Z)_{\infty, \hat{\theta}} = \mathbb{E}_{P_X Z} T_{\hat{\theta}(X, Z)} - \mathbb{E}_{P_X} \mathbb{E}_{P_Z} e^{T_{\hat{\theta}(X, Z)}} + 1
\]

\[
\leq \sup_{\theta \in \Theta} \mathbb{E}_{P_X Z} T_{\theta}(X, Z) - \mathbb{E}_{P_X} \mathbb{E}_{P_Z} e^{T_{\theta}(X, Z)} + 1 \leq I(X; Z)
\]

(7)

We now derive the sample complexity of DEMINE defined as the number of samples \( n \) required for \( \hat{I}(X, Z)_{\hat{\theta}, n} \) to be a good approximation to \( \hat{I}(X, Z)_{\infty, \hat{\theta}} \) in Theorem 1.

**Theorem 1.** For \( T_{\hat{\theta}}(X, Z) \) bounded by \( [L, U] \), given any accuracy \( \epsilon \) and confidence \( \delta \), we have:

\[
\Pr(|\hat{I}(X, Z)_{\hat{\theta}, n} - \hat{I}(X, Z)_{\infty, \hat{\theta}}| \leq \epsilon) \geq 1 - \delta
\]

when the number of validation samples \( n \) satisfies:

\[
n \geq n^*, s.t. f(n^*) \equiv \min_{0 \leq \xi \leq \epsilon} \left( 2e^{-2\xi^2n^*} + 4e^{-(\xi-\epsilon)^2n^*} \right)^{1/2} = \delta
\]

(8)
Proof. Since $T_\theta(x, z)$ is bounded by $[L, U]$, applying the Hoeffding inequality to the first half of Eq.5 yields:
\[
\Pr\left(\frac{1}{n} \sum_{i=1}^{n} T_\theta(x_i, z_i) - \mathbb{E}_{P_X} T_\theta(x, z) \geq \xi \right) \leq 2e^{-\frac{2\xi^2 n}{(L - U)^2}}.
\]
As $e^{T_\theta(x,z)}$ is bounded by $[e^L, e^U]$, applying the Hoeffding inequality twice to the second half of Eq.5:
\[
\begin{align*}
&\Pr\left(\mathbb{E}_{P_X} \mathbb{E}_{P_Z} e^{T_\theta(x,z)} - \frac{1}{n} \sum_{i=1}^{n} T_\theta(x_i, z_i) \geq \xi \right) \\
&\leq 2e^{-\frac{2\xi^2 n}{2(L - U)^2}},
\end{align*}
\]
\[
\begin{align*}
&\Pr\left(\mathbb{E}_{P_X} \mathbb{E}_{P_Z} e^{T_\theta(x,z)} - \frac{1}{n} \sum_{i=1}^{n} \frac{1}{n} \sum_{i=1}^{n} e^{T_\theta(x_i, z_j)} \geq \xi \right) \\
&\leq 2e^{-\frac{2(\xi + \zeta)^2 n}{(L - U)^2}}.
\end{align*}
\]
Combining the above bounds results in:
\[
\Pr\left(\widehat{I}(X, Z)_{n, \hat{\theta}} - \widehat{I}(X, Z)_{\infty, \hat{\theta}} \leq \xi + 2\zeta \right) \geq 1 - 2e^{-\frac{2\xi^2 n}{(L - U)^2}} - 4e^{-\frac{2(\xi + \zeta)^2 n}{(L - U)^2}}.
\]
By solving $\xi$ to minimize $n$ according to Eq.8, we have:
\[
\Pr\left(\widehat{I}(X, Z)_{n, \hat{\theta}} - \widehat{I}(X, Z)_{\infty, \hat{\theta}} \leq \epsilon \right) \geq 1 - \delta.
\]

Theorem 1 also implies the following MI lower confidence interval under limited number of samples:
\[
\Pr(I(X; Z) \geq \widehat{I}(X, Z)_{n, \hat{\theta}} - \epsilon) \geq 1 - \delta.
\]

Compared to MINE, as per the example shown in Section 3 for $M = 1$ (i.e. $L = -1$ and $U = 1$), $\delta = 0.05$, $\epsilon = 0.1$, our estimator requires $n = 10, 742$ compared to MINE requiring $n = 18, 756$, 256 i.i.d. validation samples to estimate a lower bound, which makes MI-based dependency analysis feasible for domains where data collection is prohibitively expensive, e.g. fMRI scans. In practice, sample complexity can be further optimized by optimizing hyperparameters $U$ and $L$.

Note that unlike Eq.3, Theorem 1 bounds the closeness of the DEMINE estimate, $\widehat{I}(X, Z)_{n, \hat{\theta}}$, not towards the MI lower bound $\sup_{\theta \in \Theta} \overline{I}_{MINE, e}$, but towards the MI lower bound $\overline{I}(X, Z)_{\infty, \hat{\theta}}$. Therefore, the sample complexity of DEMINE as in Eq.8 makes fair comparison with the sample complexity of MINE as in Eq.2. MINE’s higher sample complexity stems from the need to bound the generalization error of $T_\theta(x, z)$ on unseen $(x, z)$. Existing generalization bounds are known to be overly loose, as over-parameterized neural networks have been shown to generalize well in classification and regression tasks (Zhang et al., 2016). By using a learning-based formulation, DEMINE not only avoids the need to bound generalization error, but also allows further generalization improvements by learning $\theta$ through meta-learning.

In the following section, we present a meta-learning formulation, Meta-DEMINE, that learns $\hat{\theta}$ for generalization given the same model class and training samples.

4.2 Meta-Learning

Given training data $\{(x_i, z_i)_{\text{train}}, i = 1, \ldots, m\}$, Meta-DEMINE first generates MI estimation tasks each consisting of a meta-training split $A$ and a meta-val split $B$ through a novel task augmentation process. And then a parameter initialization $\theta_{\text{init}}$ is then learned to maximize MI estimation performance on the generated tasks using initialization $\theta_{\text{init}}$ as shown in Eq.9
\[
\theta_{\text{init}} = \arg \min_{\theta^{(0)} \in \Theta} \mathbb{E}_{(A, B) \in \mathcal{T}} \mathcal{L}(x, z; \theta^{(t)}), \text{with } \theta^{(t)} \equiv \text{MetaTrain}(x, z; \theta^{(0)}).
\]

Here $\theta^{(t)} = \text{MetaTrain}(x, z; \theta^{(0)})$ is the meta-training process of starting from an initialization $\theta^{(0)}$ and applying Stochastic Gradient Descent (SGD) over $t$ steps to learn $\theta$ where in every meta training iteration we have:
\[
\theta^{(t)} \leftarrow \theta^{(t-1)} - \gamma \nabla \mathcal{L}(x, z; \theta^{(t-1)}),
\]

---

1In practice, the Adam optimizer (Kingma & Ba, 2014) is used for faster optimization. The Adam optimizer uses first and second order momentum of the gradient to speed up optimization. Illustrating SGD for simplicity.
Finally, $\tilde{\theta}$ is learned using the entire training set $\{(x_i, z_i)_{train}, i = 1, \ldots, m\}$ with $\theta_{init}$ as initialization:

$$\tilde{\theta} = \text{MetaTrain}(\langle x, z \rangle_{train}, \theta_{init}).$$

**Task Augmentation**: Meta-DEMINE adapts MAML (Finn et al. [2017]) for MI lower bound maximization. MAML has been shown to improve generalization performance in N-class K-shot image classification. MI estimation, however, does not come with predefined classes and tasks. A naive approach to produce tasks would be through cross-validation – partitioning training data into meta-training and meta-validation splits. However, merely using cross-validation tasks is prone to overfitting – a $\theta_{init}$, which memorizes all training samples would as a result have memorized all meta-validation splits. Instead, Meta-DEMINE generates tasks by augmenting the cross-validation tasks through *task augmentation*. Training samples are first split into meta-training and meta-validation splits, and then transformed using the same random invertible transformation to increase task diversity. Meta-DEMINE generates invertible transformation by sequentially composing the following functions:

- **Mirror**: $m(x) = (2n - 1) x$, $n \sim \text{Bernoulli}(\frac{1}{2})$.
- **Permute**: $P(x) = x_{\pi}$, Permute dimensions.
- **Offset**: $O(x) = x + \epsilon$, $\epsilon \sim \mathcal{U}(-0.1, 0.1)$.
- **Gamma**: $G(x) = \text{sign}(x) |x|^{\gamma}$, $\gamma \sim \mathcal{U}(0.5, 2)$.

Since the MI between two random variables is invariant to invertible transformations on each variable, MetaTrain$(\cdot, \cdot)$ is expected to arrive at the same MI lower bound estimation regardless of the transformation applied. At the same time, memorization is greatly suppressed, as the same pair $(x, z)$ can have different $\log \frac{p(x, z)}{p(x)p(z)}$ under different transformations. More sophisticated invertible transformations (affine, piece-wise linear) can also be added. Task augmentation is an orthogonal approach to data augmentation. Using image classification as an example, data augmentation generates variations of the image, translated, or rotated images assuming that they are valid examples of the class. Task augmentation on the other hand, does not make such an assumption. Task augmentation requires the initial parameters $\theta_{init}$ to be capable of recognizing the same class in a world where all images are translated and/or rotated, with the assumption that the optimal initialization should easily adapt to both the upright world and the translated and/or rotated world.

**Optimization**: Solving $\theta_{init}$ using the meta-learning formulation Eq.9 poses a challenging optimization problem. The commonly used approach is back propagation through time (BPTT) which computes second order gradients and directly back propagates gradients from MetaTrain$((x, z)_{\lambda}, \theta^{(0)})$ to $\theta_{init}$. BPTT is very effective for a small number of optimization steps, but is vulnerable to exploding gradients and is memory intensive. In addition to BPTT, we find that stochastic finite difference algorithms such as Evolution Strategies (ES) (Salimans et al. [2017]) and Parameter-Exploring Policy Gradients (PEPG) (Sehnke et al. [2010]) can sometimes improve optimization robustness. In practice, we switch between BPTT and PEPG depending on the number of meta-training iterations. Meta-DEMINE algorithm is specified in Algorithm 2.

5 **Evaluation on Synthetic Datasets**

**Dataset**. We evaluate our approaches DEMINE and Meta-DEMINE against baselines and state-of-the-art approaches on 3 synthetic datasets: 1D Gaussian, 20D Gaussian and sine wave. For 1D and 20D Gaussian datasets, following Belghazi et al. [2018], we define two $k$-dimensional multivariate Gaussian random variables $X$ and $Z$ which have component-wise correlation $\text{corr}(X_i, Z_j) = \delta_{ij}\rho$, where $\rho \in (-1, 1)$ and $\delta_{ij}$ is Kronecker’s delta. Mutual information $I(X; Z)$ has a closed form solution $I(X; Z) = -k \ln(1 - \rho^2)$. For the sine wave dataset, we define two random variables $X$ and $Z$, where $X \sim \mathcal{U}(-1, 1)$, $Z = \sin(aX + \frac{\pi}{2}) + 0.05\epsilon$, and $\epsilon \sim \mathcal{N}(0, 1)$. Estimating mutual information accurately given few pairs of $(X, Z)$ requires the ability to extrapolate the sine wave given few examples. Ground truth MI for sine wave dataset is approximated by running the KSG Estimator (Kraskov et al. [2004]) on 1,000,000 samples.

**Implementation**. We compare our estimators, DEMINE and Meta-DEMINE, against the KSG estimator (Kraskov et al. [2004]), MI-KSG and MINE-f (Belghazi et al. [2018]). For both DEMINE and Meta-DEMINE, we study variance reduction mode, referred to as -vr, where hyperparameters are selected by optimizing 95% confident estimation mean ($\mu - 2\sigma_\mu$) and statistical significance mode, referred to as -sig, where hyperparameters are selected by optimizing 95% confident MI.
Algorithm 2 Meta-DEMINE

Input Data: \( \{(x, z)_{\text{train}}, (x, z)_{\text{val}}\} \)

Parameters: batch \( B \), Meta Learning Iterations \( N_M \), Task Augmentation Iterations \( N_T \), Optimization Iterations \( N_O \), Ratio \( r \), Learning rate \( \eta \), Meta Learning Rate \( \eta_{\text{meta}} \)

Output: \( \text{MI}, T_{\theta_{\text{init}}}(X, Z), T_{\theta}(X, Z) \)

1: for \( i = 1 : N_M \) do
2:   for \( j = 1 : N_T \) do
3:     \( A = r \times \text{train}, B = \text{train} - A \)
4:     Split \( (x, z)_{\text{train}} \) into \( (x, z)_A \) and \( (x, z)_B \)
5:     Transformation \( R_x \) for \( x \), \( R_z(\cdot) = \text{m}(P(O(G(\cdot)))) \)
6:     Transformation \( R_z \) for \( z \), \( R_z(\cdot) = \text{m}(P(O(G(\cdot)))) \)
7:     Compute \( \theta^{(0)}_{\text{meta}} \leftarrow \theta_{\text{init}} \)
8:   for \( k = 1 : N_O \) do
9:     Sample a batch of \( (x, z)_B \sim (x, z)_A \)
10:    Compute \( \mathcal{L}(\{R_x(x), R_z(z)\}_B, \theta^{(k)}_{\text{meta}}) \)
11:    Compute \( \nabla \theta_{\text{meta}} \mathcal{L} - \text{gradient for } \theta_{\text{meta}} \)
12:    Update \( \theta_{\text{meta}} \) using Adam \cite{Kingma2014} with \( \eta \)
13: for end
14:    Compute \( \mathcal{L}_{\text{meta}}(\{R_x(x), R_z(z)\}_B, \theta^{(N_O)}_{\text{meta}}) \)
15:    Compute \( \nabla \theta \mathcal{L}_{\text{meta}} - \text{gradient to } \theta_{\text{init}} \) using BPTT
16: end for
17: Update \( \theta_{\text{init}} \) using Adam \cite{Kingma2014} with \( \eta_{\text{meta}} \)
18: end for
19: \( \theta^{(0)} \leftarrow \theta_{\text{init}} \)
20: for \( i = 1 : N_O \) do
21:    Sample a batch of \( (x, z)_B \sim (x, z)_{\text{train}} \)
22:    Compute \( \mathcal{L}(\{x, z\}_B, \theta^{(i)}) \)
23:    Compute gradient \( \nabla \theta \mathcal{L} \)
24:    Update \( \theta \) using Adam with \( \eta \)
25: end for
26: Compute \( \text{MI} = \mathcal{L}(\{(x, z)_{\text{val}}, \theta^{(N_O)}\}) \)
27: return \( \text{MI}, \theta_{\text{init}}, \theta^{(N_O)} \)

lower bound \((\mu - \epsilon)\). Samples \((x, z)\) are split 50%-50% into \((x, z)_{\text{train}}\) and \((x, z)_{\text{val}}\). We use a separable network architecture \( T_\theta(x, z) = M(\text{tanh}(w \cos(f(x), g(z)) + b) - t) \). \( f \) and \( g \) are MLP encoders that embed signals \( x \) and \( z \) into vector embeddings. Hyperparameters \( t \in [-1, 1] \) and \( M \) control upper and lower bounds \( T_\theta(x, z) \in [-M(1 + t), M(1 - t)] \). Parameters \( w \) and \( b \) are learnable parameters. MLP design and optimization hyperparameters are selected using Bayesian hyperparameter optimization \cite{Bergstra2013} described below.

Hyperparameter search on DEMINE-vr and DEMINE-sig was conducted using the hyperopt package.\footnote{Hyperopt package: \url{https://github.com/hyperopt/hyperopt}} Seven hyperparameters were involved in hyperparameter search: 1) number of encoder layers \([1, 5]\), 2) encoder hidden size \([8, 256]\), 3) learning rate \( \eta [10^{-4}, 3 \times 10^{-1}] \) in log scale, 4) number of optimization iterations \( N_O [5, 200] \) (sine wave \([5, 5000]\)) in log scale, 5) batch size \( B [256, 1024]\), 6) \( M, [10^{-3}, 5] \) in log scale, 7) \( t, [-1, 1] \). Mean \( \mu \) and sample standard deviation \( \sigma \) of MI estimate computed over 3-fold cross-validation on \((x, z)_{\text{train}}\). DEMINE-vr maximizes two sigma low \( \mu - 2\sigma \mu \) where \( \sigma_{\mu} = \frac{1}{\sqrt{N}} \sigma \) due to 3-fold cross-validation. DEMINE-sig maximizes statistical significance \( \mu - \epsilon \) where \( \epsilon \) is two-sided 95% confidence interval of MI. Meta-DEMINE-vr and Meta-DEMINE-sig subsequently reuse these hyperparameters as DEMINE-vr and DEMINE-sig.

Meta-learning hyperparameters are chosen as outer loop \( N_M = 3, 000 \) iterations, task augmentation \( N_T = 1 \) iterations, \( r = 0.8, \eta_{\text{meta}} = \frac{3}{4} \), with task augmentation mode \( m_i(P(O(\cdot))) \). \( N_O \) was capped at 30 iterations for 1D and 2D Gaussian datasets due to memory limit. For the sine wave datasets with large \( N_O \), we used PEPG \cite{Sehnke2010} rather than BPTT.
For MI-KSG, we use off-the-shelf implementation by [Gao et al., 2017] with default number of
nearest neighbors $k = 3$. MI-KSG does not provide any confidence interval. For MINE-f, we use the
same network architecture same as DEMINE-vr. We implement both the original formulation which
optimizes $T_{θ}$ on $(x, z)$ till convergence (10k iters), as well as our own implementation MINE-f-ES
with early stopping, where optimization is stopped after the same number of iterations as DEMINE-
vr to control overfitting.

**Results.** Figure 1(a) shows MI estimation performance on 20D Gaussian datasets with varying
$ρ \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ using $N = 300$ samples. Results are averaged over 5 runs to compare
estimator bias, variance and confidence. Note that Meta-DEMINE-sig detects the highest $ρ < 0.05
confidence MI, outperforming DEMINE-sig which is a close second. Both detect $ρ < 0.05$ statistically
significant dependency starting $ρ = 0.3$, whereas estimations of all other approaches are low
confidence. It shows that in contrary to common belief, estimating the variational lower bounds with
high confidence can be challenging under limited data. MINE-f estimates positive MI when $ρ = 0$, both due to overfitting, despite MINE-f-ES having the lowest empirical bias. DEMINE variants have relatively high empirical bias but low variance due to tight
upper and lower bound control, which provides a different angle to understand bias-variance trade
off in MI estimation (Poole et al., 2018).

Figure 1(b,c,d) shows MI estimation performance on 1D, 20D Gaussian and sine wave datasets with
fixed $ρ = 0.8, 0.3$ and $α = 8π$ respectively, with varying $N \in \{30, 100, 300, 1000, 3000\}$ number of
samples. More samples asymptotically improves empirical bias across all estimators. As opposed to
1D Gaussian datasets which are well solved by $N = 300$ samples, higher-dimensional 20D Gaussian
and higher-complexity sine wave datasets are much more challenging and are not solved using $N =
3000$ samples with a signal-agnostic MLP architecture. DEMINE-sig and Meta-DEMINE-sig detect
$ρ < 0.05$ statistically significant dependency on not only 1D and 20D Gaussian datasets where $x$
and $z$ have non-zero correlation, but also on the sine wave datasets where correlation between $x$ and
$z$ is 0. This means that DEMINE-sig and Meta-DEMINE-sig can be used for nonlinear dependency
testing to complement linear correlation testing.

We study the effect of cross-validation meta-learning and task augmentation on 20D Gaussian with
$ρ = 0.3$ and $N = 300$. Figure 2 plots performance of Meta-DEMINE-vr over $N_M = 3000$
meta iterations under combinations of task augmentations modes and number of adaptation iterations
$N_O \in \{0, 20\}$. Overall, task augmentation modes which involve axis flipping $m(\cdot)$ and
permutation $P(\cdot)$ are the most successful. With $N_O = 20$ steps of adaptation, task augmentation
modes $P(\cdot), m(P(\cdot))$ and $m(P(O(\cdot)))$ prevent overfitting and improves performance. The
performance improvements of task augmentation is not simply from change in batch size, learning rate
or number of optimization iterations, because meta-learning without task augmentation for both
$N_O = 0$ and 20 could not outperform baseline. Meta-learning without task augmentation and with
task augmentation but using only $O(\cdot)$ or $G(\cdot)$ result in overfitting. Task augmentation with $m(\cdot)$ or
$m(P(O(G(\cdot))))$ prevent overfitting, but do not provide performance benefits, possibly because
their complexity is insufficient or excessive for 20 adaptation steps. Further more, task augmentation with
no adaptation ($N_O = 0$) falls back to data augmentation, where samples from transformed distributions
are directly used to learn $T_{θ}(x, z)$. Data augmentation with $O(\cdot)$ outperforms no augmentation,
but is unable to outperform baseline and suffers from overfitting. It shows that task augmentation
provides improvements orthogonal to data augmentation.

6 APPLICATION: fMRI INTER-SUBJECT CORRELATION (ISC) ANALYSIS

Humans use language to effectively transmit brain representations among conspecifics. For example,
after witnessing an event in the world, a speaker may use verbal communication to evoke neural
representations reflecting that event in a listener’s brain (Hasson et al., 2012). The efficacy of this
transmission, in terms of listener comprehension, is predicted by speaker-listener neural synchrony and
synchrony among listeners (Stephens et al., 2010). To date, most work has measured brain-
to-brain synchrony by locating statistically significant inter-subject correlation (ISC), quantified as the
Pearson product-moment correlation coefficient between response time series for corresponding
volumes or regions of interest (ROIs) across individuals (Hasson et al., 2004; Schippers et al., 2010;
Silbert et al., 2014; Nastase et al., 2019). Using DEMINE and Meta-DEMINE for statistical depen-
dency testing, we can extend ISC analysis to capture nonlinear and higher-order interactions in
continuous fMRI responses. Specifically, given synchronized fMRI response frames in two brain
Under review as a conference paper at ICLR 2020

(a) 20D Gaussian dataset, $N = 300$ samples

(b) 1D Gaussian dataset, $\rho = 0.8$

(c) 20D Gaussian dataset, $\rho = 0.3$

(d) Sine wave dataset, $a = 8\pi$

Figure 1: Comparing MI Estimation performance of DEMINE and Meta-DEMINE with the KSG estimator [Kraskov et al., 2004] and MINE-f [Belghazi et al., 2018] on different datasets using varying number of samples. The bars show estimator mean and standard deviation averaged over 5 runs with different seeds. The error bars show 95% confidence interval (not available for MI-KSG). The statistical significance focused variants DEMINE-sig and Meta-DEMINE-sig achieves the highest 95% confident MI estimation. Meta-DEMINE improves over DEMINE most of the time.

(1) Meta-DEMINE-vr $N_O = 0$.

(2) Meta-DEMINE-vr $N_O = 10$.


Figure 2: To study the effect of task augmentation and number of adaptation steps, we run Meta-DEMINE-vr with different task augmentation modes and vary number of adaptation iterations $N_O \in \{0, 10, 20\}$ on Gaussian 20D, $\rho = 0.3$ dataset. Combinations of permutation and mirroring operations are effective in reducing overfitting and improving performance.

Table 1: Number of HCP-MMP1 regions with significant correlation ($r$) and MI (DEMINE, Meta-DEMINE) during listening.

<table>
<thead>
<tr>
<th>No. shared</th>
<th>$r$ DEMINE-sig</th>
<th>Meta-DEMINE-sig</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r$</td>
<td>37</td>
<td>24</td>
</tr>
<tr>
<td>DEMINE-sig</td>
<td>24</td>
<td>28</td>
</tr>
<tr>
<td>Meta-DEMINE-sig</td>
<td>23</td>
<td>26</td>
</tr>
</tbody>
</table>

Table 2: Segment classification accuracy for NeuralMI versus Pearson’s correlation in 1-vs-rest*.

<table>
<thead>
<tr>
<th>Classification</th>
<th>ISC Mask</th>
<th>dDMN Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>F</td>
</tr>
<tr>
<td>Chance</td>
<td>3.7</td>
<td>1.8</td>
</tr>
<tr>
<td>Pearson’s r</td>
<td>35.0</td>
<td>20.4</td>
</tr>
<tr>
<td>DEMINE-vr 1vR</td>
<td>42.8</td>
<td>28.0</td>
</tr>
<tr>
<td>Meta-DEMINE-vr 1vR</td>
<td>47.2</td>
<td>32.5</td>
</tr>
</tbody>
</table>

Abbreviations: P: Pieman; F: Forgot; Br: Bronx; Bk: Black; MI: Mutual Information.

*Note that all the results are averaging over the subjects.
regions $X$ and $Z$ across $K$ subjects $X_i, Z_i, i = 1, \ldots, K$ as random variables. We model the conditional mutual information $I(X_i; Z_j | i \neq j)$ as the MI form of pair-wise ISC analysis. By definition, $I(X_i; Z_j | i \neq j)$ first computes MI between activations $X_i$ and $Z_j$ from subjects $i$ and $j$ respectively, and then average across pairs of subjects $i \neq j$. It can be lower bounded using Eq. \[7\] by learning a $T_{\theta}(x, z)$ shared across all subject pairs.

**Dataset.** We study MI-based and correlation-based ISC on a fMRI story comprehension dataset by Nastase et al. (2019) with 40 participants listening to four spoken stories. Average story duration is 11 minutes. An fMRI frame with full brain coverage is captured at repetition time 1 TR = 1.5 seconds with a 2.5mm isotropic spatial resolution. We restricted our analysis to subsets of voxels defined using independent data from previous studies: functionally-defined masks of high ISC voxels (ISC; 3,800 voxels) and dorsal Default-Mode Network voxels (dDMN; 3,940 voxels) from Simony et al. (2016) as well as 180 HCP-MMP1 multimodal cortex parcels from Glasser et al. (2016). All masks were defined in MNI space.

**Implementation.** We compare MI-based ISC using DEMINE and Meta-DEMINE with correlation-based ISC using Pearson’s correlation. DEMINE and Meta-DEMINE setup follows Section Section [5]. The fMRI data were partitioned by subject into a train set of 20 subjects and a validation set of 20 different subjects. Residual 1D CNN is used instead of MLP as the encoder for studying temporal dependency. For Pearson’s correlation, high-dimensional signals are reshaped to 1D for correlation analysis.

**Results.** We first examine, for the fine grained HCM-MMP1 brain regions, which have $p < 0.05$ statistically significant MI and Pearson’s correlation. Table [1] shows the result. Overall, more regions have statistically significant correlation than dependency. This is expected because correlation requires less data to detect. But Meta-DEMINE is able to find 6 brain regions that have statistically significant dependency but lacks significant correlation. This shows that MI analysis can be used to complement correlation-based ISC analysis.

By considering temporal ISC over time, fMRI signals can be modeled with improved accuracy. In Table [2] we apply DEMINE and Meta-DEMINE with $L = 10$ TRs (15s) sliding windows as random variables to study amount of information that can be extracted from ISC and dDMN masks. We use between-subject time-segment classification (BSC) for evaluation (Haxby et al., 2011; Guntupalli et al., 2016). Each fMRI scan is divided into $K$ non-overlapping $L = 10$ TRs time segments. The BSC task is one versus rest retrieval: retrieve the corresponding time segment $z$ of an individual given a group of time segments $x$ excluding that individual, measured by top-1 accuracy. For retrieval score, $T_{\theta}(X, Z)$ is used for DEMINE and Meta-DEMINE and $\rho(X, Z)$ is used for Pearson’s correlation as a simple baseline. With CNN as encoder, DEMINE and Meta-DEMINE model the signal better and achieve higher accuracy. Also, Meta-DEMINE is able to extract 0.75 nats of MI from the ISC mask over 10 TRs or 15s, which could potentially be improved by more samples.

7 CONCLUSION

We illustrated that a predictive view of the MI lower bounds coupled with meta-learning results in data-efficient variational MI estimators, DEMINE and Meta-DEMINE, that are capable of performing statistical test of dependency. We also showed that our proposed task augmentation reduces overfitting and improves generalization in meta-learning. We successfully applied MI estimation to real world, data-scarce, fMRI datasets. Our results suggest a greater avenue of using neural networks and meta-learning to improve MI analysis and applying neural network-based information theory tools to enhance the analysis of information processing in the brain. Model-agnostic, high-confidence, MI lower bound estimation approaches – including MINE, DEMINE and Meta-DEMINE – are limited to estimating small MI lower bounds up to $O(\log N)$ as pointed out in McAllester & Statos (2018), where $N$ is the number of samples. In real fMRI datasets, however, strong dependency is rare and existing MI estimation tools are limited more by their ability to accurately characterize the dependency. Nevertheless, when quantitatively measuring strong dependency, cross-entropy (McAllester & Statos 2018) or model-based quantities, alternatives to MI, such as correlation or CCA, may be measured with high confidence.
REFERENCES


Carol Daniel. I knew you were black. [https://themoth.org/stories/1-knew-you-were-black](https://themoth.org/stories/1-knew-you-were-black), 2018. Accessed: 2018-10-12.


A  APPENDIX

Additional Details of the fMRI Dataset  The dataset we used contains 40 participants (mean age = 23.3 years, standard deviation = 8.9, range: 18-53; 27 female) recruited to listen to four spoken stories. The stories were renditions of “Pie Man” and “Running from the Bronx” by Jim O’Grady (O’Grady, 2018a,b), “The Man Who Forgot Ray Bradbury” by Neil Gaiman (Gaiman, 2018), and “I Knew You Were Black” by Carol Daniel (Daniel, 2018); story durations were 7, 9, 14, and 13 minutes, respectively. After scanning, participants completed a questionnaire comprising 25-30 questions per story intended to measure narrative comprehension. The questionnaires included multiple choice, True/False, and fill-in-the-blank questions, as well as four additional subjective ratings per story. Functional and structural images were acquired using a 3T Siemens Prisma with a 64-channel head coil. Briefly, functional images were acquired in an interleaved fashion using gradient-echo echo-planar imaging with a multiband acceleration factor of 3 (TR/TE = 1500/31 ms where TE stands for “echo time”, resolution = 2.5 mm isotropic voxels, full brain coverage).

All fMRI data were formatted according to the Brain Imaging Data Structure (BIDS) standard (Gorgolewski et al., 2016) and preprocessed using the fMRIPrep library (Esteban et al., 2018). Functional data were corrected for slice timing, head motion, and susceptibility distortion, and normalized to MNI space using nonlinear registration. Nuisance variables comprising head motion parameters, framewise displacement, linear and quadratic trends, sine/cosine bases for high-pass filtering (0.007 Hz), and six principal component time series from cerebrospinal fluid (CSF) and white matter (WM) were regressed out of the signal using the Analysis of Functional NeuroImages (AFNI) software suite (Cox, 1996).

The fMRI data comprise $X \in \mathbb{R}^{V \times T}$ for each subject, where $V_i$ represents the flattened and masked voxel space and $T$ represents the number of samples (in TRs) during auditory stimulus presentation.  

Additional Details on Dataset Collection  Functional and structural images were acquired using a 3T Siemens Magnetom Prisma with a 64-channel head coil. Functional, blood-oxygenation-level-dependent (BOLD) images were acquired in an interleaved fashion using gradient-echo echo-planar imaging with pre-scan normalization, fat suppression, a multiband acceleration factor of 3, and no in-plane acceleration: TR/TE = 1500/31 ms, flip angle = 67°, bandwidth = 2480 hz per pixel, resolution = 2.5 mm isotropic voxels, matrix size = 96 x 96, Field of view (FoV) = 240 x 240 mm, 48 axial slices with roughly full brain coverage and no gap, anterior-posterior phase encoding. At the beginning of each scanning session, a T1-weighted structural scan (where T1 stands for “longitudinal relaxation time”), was acquired using a high-resolution single-shot Magnetization-Prepared 180 degrees radio-frequency pulses and RApid Gradient-Echo (MPRAGE) sequence with an in-plane acceleration factor of 2 using GeneRalized Autocalibrating Partial Parallel Acquisition (GRAPPA): TR/TE/TI = 2530/3.3/1100 ms where TI stands for inversion time, flip angle = 7°, resolution = 1.0 x 1.0 x 1.0 mm voxels, matrix size = 256 x 256, FoV = 256 x 256 x 176 mm, 176 sagittal slices, ascending acquisition, anterior-posterior phase encoding, no fat suppression, 5 min 53 s total acquisition time. At the end of each scanning session a T2-weighted structural scan was acquired using the same acquisition parameters and geometry as the T1-weighted structural image: TR/TE = 3200/428 ms, 4 minutes 40 seconds total acquisition time. A field map was acquired at the beginning of each scanning session, but was not used in subsequent analyses.

Additional Details on Dataset Preprocessing  Preprocessing was performed using the fMRIPrep library (Esteban et al., 2018), a Nipype library (Gorgolewski et al., 2011) based tool. T1-weighted images were corrected for intensity non-uniformity using the N4 bias field correction algorithm (Tustison et al., 2010) and skull-stripped using Advanced Normalization Tools (ANTs) (Avants et al., 2008). Nonlinear spatial normalization to the International Consortium for Brain Mapping (ICBM) T152 Nonlinear Asymmetrical template version 2009c (Fonov et al., 2009) was performed using ANTs. Brain tissue segmentation cerebrospinal fluid, white matter, and gray matter was

---

5Two of the stories were told by a professional storyteller undergoing an fMRI scan; however, fMRI data for the speaker were not analyzed for the present work due to the head motion induced by speech production.

6The study was conducted in compliance with the Institutional Review Board of the University

https://github.com/poldracklab/fmriprep
https://github.com/nipy/nipype
was performed using FSL library’s \texttt{FAST} tool \cite{Zhang2001}. Functional images were slice timing corrected using AFNI software’s \texttt{3dTshift} \cite{Cox1996} and corrected for head motion using FSL library’s \texttt{MCFLIRT} tool \cite{Jenkinson2002}. “Fieldmap-less” distortion correction was performed by co-registering each subject’s functional image to that subject’s intensity-inverted T1-weighted image \cite{Wang2017} constrained with an average field map template \cite{Treibert2016}. This was followed by co-registration to the corresponding T1-weighted image using FreeSurfer software’s \texttt{boundary-based registration} \cite{Greve2009} with 9 degrees of freedom. Motion correcting transformations, field distortion correcting warp, BOLD-to-T1 transformation and T1-to-template (MNI) warp were concatenated and applied in a single step with Lanczos interpolation using ANTs. Physiological noise regressors were extracted applying “a Component Based Noise Correction Method” \texttt{aCompCor} \cite{Behzadi2007}. Six principal component time series were calculated within the intersection of the subcortical mask and the union of CSF and WM masks calculated in T1w (T1 weighted) space, after their projection to the native space of each functional run. Framewise displacement \cite{Power2014} was calculated for each functional run. Functional images were downsampled to 3 mm resolution. Nuisance variables comprising six head motion parameters (and their derivatives), framewise displacement, linear and quadratic trends, sine/cosine bases for high-pass filtering (0.007 Hz cutoff), and six principal component time series from an anatomically-defined mask of cerebrospinal fluid and white matter were regressed out of the signal using AFNI’s \texttt{3dTproject} \cite{Cox1996}. Functional response time series were z-scored for each voxel.