PNAS RESEARCH ARTICLE | STATISTICS

Cooperative learning for multiview analysis

Daisy Yi Ding[a], Shuangning Li[b], Balasubramanian Narasimhan[a,b], and Robert Tibshirani[a,b,1]

Contributed by Robert Tibshirani; received February 4, 2022; accepted August 9, 2022; reviewed by Adam Olshen and Ji Zhu

We propose a method for supervised learning with multiple sets of features ("views"). The multiview problem is especially important in biology and medicine, where "-omics" data, such as genomics, proteomics, and radiomics, are measured on a common set of samples. "Cooperative learning" combines the usual squared-error loss of predictions with an "agreement" penalty to encourage the predictions from different data views to agree. By varying the weight of the agreement penalty, we get a continuum of solutions that include the well-known early and late fusion approaches. Cooperative learning chooses the degree of agreement (or fusion) in an adaptive manner, using a validation set or cross-validation to estimate test set prediction error. One version of our fitting procedure is modular, where one can choose different fitting mechanisms (e.g., lasso, random forests, boosting, or neural networks) appropriate for different data views. In the setting of cooperative regularized linear regression, the method combines the lasso penalty with the agreement penalty, yielding feature sparsity. The method can be especially powerful when the different data views share some underlying relationship in their signals that can be exploited to boost the signals. We show that cooperative learning achieves higher predictive accuracy on simulated data and real multiomics examples of labor-onset prediction. By leveraging aligned signals and allowing flexible fitting mechanisms for different modalities, cooperative learning offers a powerful approach to multiomics data fusion.

data fusion | multiomics | supervised learning | sparsity

Significance

Multiview analysis with "-omics" data, such as genomics and proteomics, measured on a common set of samples represents an increasingly important challenge in biology and medicine. Commonly used approaches can be broadly categorized into early and late fusion, depending on when "fusion" occurs. We introduce a supervised learning algorithm—"cooperative learning"—that encompasses both early and late fusion and blended versions of these methods. This algorithm encourages the predictions from different views to agree and chooses the degree of agreement in a data-adaptive manner. By leveraging aligned signals in multiomics, it can yield better predictions on tasks such as disease classification and treatment response prediction and has implications for improving diagnostics and therapeutics.

With new technologies in biomedicine, we are able to generate and collect data of various modalities , including genomics, epigenomics, transcriptomics, proteomics, and metabolomics (Fig. 1*A*). Integrating heterogeneous features on a common set of observations provides a unique opportunity to gain a comprehensive understanding of an outcome of interest. It offers the potential for making discoveries that are hidden in data analyses of a single modality and achieving more accurate predictions of the outcome (1–6). While "multiview data analysis" can mean different things, we use it here in the context of supervised learning, where the goal is to fuse different data views to model an outcome of interest.

To give a concrete example, assume that a researcher wants to predict cancer outcomes from RNA expression and DNA-methylation measurements for a set of patients. The researcher suspects that: 1) Both data views potentially have prognostic value; and 2) the two views share some underlying relationship with each other, as DNA methylation regulates gene expression and can repress the expression of tumor suppressor genes or promote the expression of oncogenes. Should the researcher use both data views for downstream prediction, or just use one view or the other? If using both views, how can the researcher leverage their underlying relationship in making more accurate predictions? Is there a way to strengthen the shared signals in the two data views, while reducing idiosyncratic noise?

There are two broad categories of existing "data fusion methods" for the multiview problem (Fig. 1*B*). They differ in the stage at which the "fusion" of predictors takes place, namely, early fusion and late fusion. Early fusion works by transforming the multiple data views into a single representation before feeding the aggregated representation into a supervised learning model of choice (7–10). The simplest approach is to column-wise concatenate the M datasets $X_1, \ldots, X_M$ to obtain a combined matrix X, which is then used as the input to a supervised learning model. Another type of early fusion approach projects each high-dimensional dataset into a low-dimensional space using methods such as principal component analysis or autoencoders (11, 12). Then, one combines the low-dimensional representations through aggregation and feeds the aggregated matrix into a supervised learning model. Early fusion approaches have an important limitation that they do not explicitly leverage the underlying relationship across data views. Late fusion, or "integration," refers to methods where individual models are first built from the distinct data views, and then the predictions of the individual models are combined into the final predictor (13–17).

In this paper, we propose a method to multiview data analysis called "cooperative learning," a supervised learning approach that fuses the different views in a systematic way.

Author affiliations: [a]Department of Biomedical Data Science, Stanford University, Stanford, CA 94305; and [b]Department of Statistics, Stanford University, Stanford, CA 94305

Author contributions: D.Y.D. and R.T. designed research; D.Y.D., S.L., B.N., and R.T. performed research; D.Y.D. analyzed data; and D.Y.D. and R.T. wrote the paper.

Reviewers: A.O., UCSF Helen Diller Family Comprehensive Cancer Center; and J.Z., University of Michigan.

The authors declare no competing interest.

Copyright © 2022 the Author(s). Published by PNAS. This article is distributed under Creative Commons Attribution-NonCommercial-NoDerivatives License 4.0 (CC BY-NC-ND).

[1]To whom correspondence may be addressed. Email: tibs@stanford.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2202113119/-/DCSupplemental.

Published September 12, 2022.

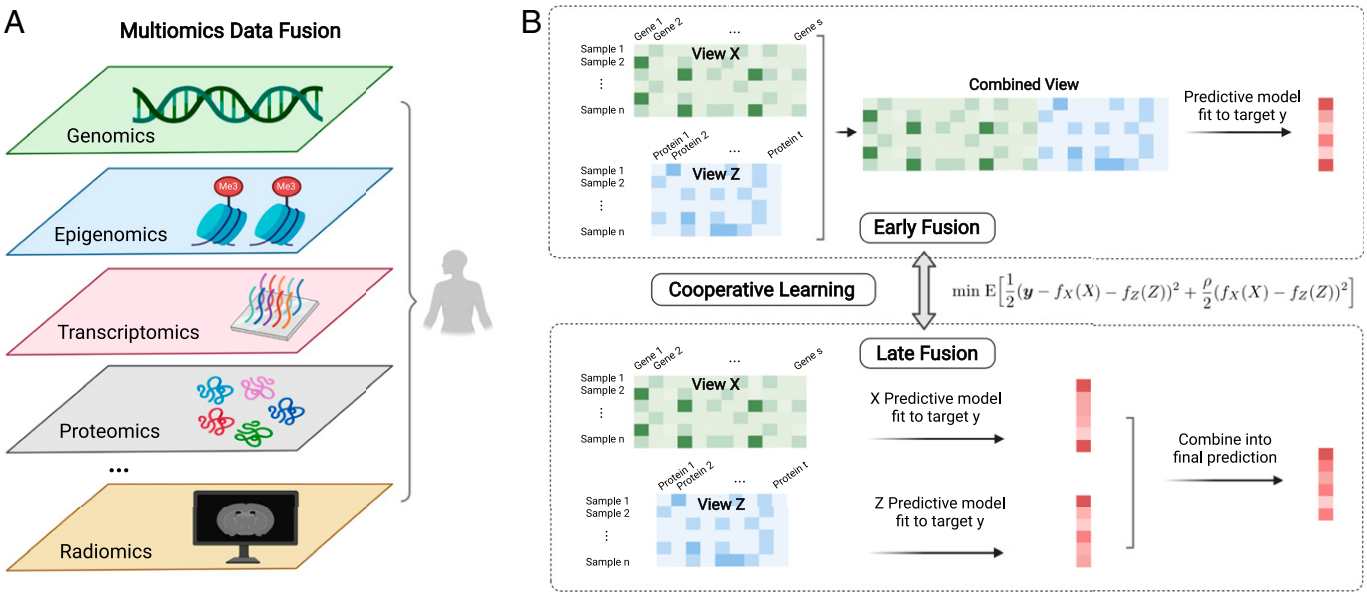

Fig. 1. Framework for multiomics data fusion. (*A*) Advances in biotechnologies have enabled the collection of a myriad of "-omics" data, ranging from genomics to proteomics, measured on a common set of samples. These data capture the molecular variations of human health at multiple levels and can help us understand complex biological systems in a more comprehensive way. Fusing the data offers the potential to improve predictive accuracy of disease phenotypes and treatment response, thus enabling better diagnostics and therapeutics. However, multiview analysis of omics data presents challenges, such as increased dimensionality, noise and complexity. (*B*) Commonly used approaches to the problem can be broadly categorized into early and late fusion. Early fusion begins by transforming all datasets into a single representation, which is then used as the input to a supervised learning model of choice. Late fusion works by developing first-level models from individual data views and then combining the predictions by training a second-level model as the final predictor. Encompassing early and late fusion, cooperative learning combines the usual squared error loss of predictions with an agreement penalty term to encourage the predictions from different data views to align.

The method combines the usual squared error loss of predictions with an "agreement" penalty that encourages the predictions from different data views to align. By varying the weight of the agreement penalty, we get a continuum of solutions that include the commonly used early and late fusion approaches. Our proposal can be especially powerful when the different data views share some underlying relationship in their signals that can be leveraged to strengthen the signals.

Cooperative Learning

Cooperative Learning with Two Data Views. We begin with a simple form of our proposal for the population (random variable) setting. Let $X \in \mathcal{R}^{n \times p_x}$, $Z \in \mathcal{R}^{n \times p_z}$—representing two data views—and $\boldsymbol{y} \in \mathcal{R}^n$ be a real-valued random variable (the target). Fixing the hyperparameter $\rho \geq 0$, we propose to minimize the population quantity:

$$\min \mathrm{E}\left[\frac{1}{2}(\boldsymbol{y} - f_X(X) - f_Z(Z))^2 + \frac{\rho}{2}(f_X(X) - f_Z(Z))^2\right]. \quad [1]$$

The first term above is the usual prediction error, while the second term is an agreement penalty, encouraging the predictions from different views to agree. This penalty term is related to "contrastive learning" (18, 19), which we discuss in more detail in *Materials and Methods*.

The solution to Eq. **1** has fixed points:

$$f_X(X) = \mathrm{E}\left[\frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)f_Z(Z)}{(1+\rho)} | X\right],$$
$$f_Z(Z) = \mathrm{E}\left[\frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)f_X(X)}{(1+\rho)} | Z\right]. \quad [2]$$

We can optimize the objective by repeatedly updating the fit for each data view in turn, holding the other view fixed. When updating a function, this approach allows us to apply the fitting method for that data view to a penalty-adjusted "partial residual." For more than two views, this generalizes easily (*Materials and Methods*).

The following relationships to early and late fusion can be seen immediately:

- If $\rho = 0$, from Eq. **1**, we see that cooperative learning chooses a functional form for f_X and f_Z and fits them together. If these functions are additive (for example, linear), then it yields a simple form of early fusion, where we simply use the combined set of features in a supervised learning procedure.
- If $\rho = 1$, then from Eq. **2**, we see that the solutions are the average of the marginal fits for X and Z. This is a simple form of late fusion.

We explore the relation of cooperative learning to early/late fusion in more detail in *Relation to Early/Late Fusion*, in the setting of regularized linear regression.

Note that this "one-at-a-time" fitting procedure is modular, so that we can choose a fitting mechanism appropriate for each data view. Specifically:

- For quantitative features like gene expression, copy number variation, or methylation: regularized regression (lasso or elastic net), a generalized additive model, boosting, random forests, or neural networks.
- For images: a convolutional neural network (CNN).
- For time-series data: an autoregressive model or a recurrent neural network.

We illustrate this on a simulated image and omics example in *Results*.

Cooperative Regularized Linear Regression. We make our proposal more concrete in the setting of cooperative regularized linear regression. Consider feature matrices $X \in \mathcal{R}^{n \times p_x}$, $Z \in \mathcal{R}^{n \times p_z}$, and our target $\boldsymbol{y} \in \mathcal{R}^n$. We assume that the columns of X and Z have been standardized, and $\boldsymbol{y}$ has mean zero (hence, we can

omit the intercept below). For a fixed value of the hyperparameter $\rho \geq 0$, we want to find $\boldsymbol{\theta_x} \in \mathcal{R}^{p_x}$ and $\boldsymbol{\theta_z} \in \mathcal{R}^{p_z}$ that minimize:

$$J(\boldsymbol{\theta_x}, \boldsymbol{\theta_z}) = \frac{1}{2}||\boldsymbol{y} - X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z}||^2 + \frac{\rho}{2}||(X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z})||^2 + \lambda_x P^x(\boldsymbol{\theta_x}) + \lambda_z P^z(\boldsymbol{\theta_z}), \quad [3]$$

where ρ is the hyperparameter that controls the relative importance of the agreement penalty term $||(X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z})||^2$ in the objective, and P^x and P^z are penalty functions. Most commonly, we use ℓ_1 penalties, giving the objective function:

$$J(\boldsymbol{\theta_x}, \boldsymbol{\theta_z}) = \frac{1}{2}||\boldsymbol{y} - X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z}||^2 + \frac{\rho}{2}||(X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z})||^2 + \lambda_x||\boldsymbol{\theta_x}||_1 + \lambda_z||\boldsymbol{\theta_z}||_1. \quad [4]$$

Note that when $\rho = 0$, this reduces to early fusion, where we simply concatenate the columns of X and Z and apply lasso. Furthermore, in *Relation to Early/Late Fusion*, we show that $\rho = 1$ yields a late fusion estimate.

In our experiments, we standardize the features and simply take $\lambda_x = \lambda_z = \lambda$. We have found that, generally, there is often no advantage to allowing different λ values for different views. However, for completeness, in *SI Appendix*, section 1, we outline an adaptive strategy for optimizing over λ_x and λ_z. We call this "adaptive cooperative learning" in our studies.

With a common λ, the objective becomes

$$J(\boldsymbol{\theta_x}, \boldsymbol{\theta_z}) = \frac{1}{2}||\boldsymbol{y} - X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z}||^2 + \frac{\rho}{2}||(X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z})||^2 + \lambda(||\boldsymbol{\theta_x}||_1 + ||\boldsymbol{\theta_z}||_1), \quad [5]$$

and we can compute a regularization path of solutions indexed by λ.

Problem [**5**] is convex, and the solution can be computed as follows. Letting

$$\tilde{X} = \begin{pmatrix} X & Z \\ -\sqrt{\rho}X & \sqrt{\rho}Z \end{pmatrix}, \tilde{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix}, \tilde{\boldsymbol{\beta}} = \begin{pmatrix} \boldsymbol{\theta_x} \\ \boldsymbol{\theta_z} \end{pmatrix}, \quad [6]$$

then the equivalent problem to Eq. **5** is

$$\frac{1}{2}||\tilde{\boldsymbol{y}} - \tilde{X}\tilde{\boldsymbol{\beta}}||^2 + \lambda(||\boldsymbol{\theta_x}||_1 + ||\boldsymbol{\theta_z}||_1). \quad [7]$$

This is a form of the lasso and can be computed, for example, by the glmnet package (20). This problem has $2n$ observations and $p_x + p_z$ features.

Let Lasso$(X, \boldsymbol{y}, \lambda)$ denote the generic problem:

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{y} - X\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1. \quad [8]$$

We outline the direct algorithm for cooperative regularized regression in Algorithm 1.

Remark A: We note that for cross-validation (CV) to estimate λ and ρ, we do not form folds from the rows of $\tilde{X}$, but, instead, form folds from the rows of X and Z and then construct the corresponding $\tilde{X}$.

Remark B: We can add ℓ_2 penalties to the objective in Eq. **5**, replacing $\lambda(||\boldsymbol{\theta_x}||_1 + ||\boldsymbol{\theta_z}||_1)$ by the elastic net form

$$\lambda\Big[(1-\alpha)(||\boldsymbol{\theta_x}||_1 + ||\boldsymbol{\theta_z}||_1) + \alpha(||\boldsymbol{\theta_x}||_2^2/2 + ||\boldsymbol{\theta_z}||_2^2/2)\Big]. \quad [9]$$

This leads to elastic net fitting, in place of the lasso, in the last step of the algorithm. This option is included in our publicly available software implementation of cooperative learning.

Algorithm 1 *Direct Algorithm for Cooperative Regularized Regression:*

Input: $X \in \mathcal{R}^{n \times p_x}$ and $Z \in \mathcal{R}^{n \times p_z}$, the response $\boldsymbol{y} \in \mathcal{R}^n$, and a grid of hyperparameter values $(\rho_{\texttt{min}}, \ldots, \rho_{\texttt{max}})$.

for $\rho \leftarrow \rho_{\texttt{min}}, \ldots, \rho_{\texttt{max}}$ **do**

Set

$$\tilde{X} = \begin{pmatrix} X & Z \\ -\sqrt{\rho}X & \sqrt{\rho}Z \end{pmatrix}, \tilde{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix}.$$

Solve Lasso$(\tilde{X}, \tilde{\boldsymbol{y}}, \lambda)$ over a decreasing grid of λ values.

end

Select the optimal value of ρ^* based on the CV error and get the final fit.

We show here an illustrative simulation study of cooperative learning in the regression setting in Fig. 2*A*. We will discuss more comprehensive studies in *Results*. In Fig. 2*A*, the first and second plots correspond to the settings where the two data views X and Z are correlated, while in the third plot, X and Z are uncorrelated. We see that when the data views are correlated, cooperative learning offers significant performance gains over the early and late fusion methods, by encouraging the predictions from different views to agree. When the data views are uncorrelated and only one view X contains signal as in the third plot, early and late fusion methods hurt performance, as compared to the separate model fit on only X, while adaptive cooperative learning is able to perform on par with the separate model.

One-at-a-Time Algorithm for Cooperative Regularized Linear Regression. As an alternative, one can optimize Eq. **4** by iteratively optimizing over $\boldsymbol{\theta_x}$ and $\boldsymbol{\theta_z}$, fixing one and optimizing over the other. The updates are as follows:

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{x}} = \text{Lasso}(X, \boldsymbol{y}_{\boldsymbol{x}}^*, \lambda_x), \text{where} \boldsymbol{y}_{\boldsymbol{x}}^* = \frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)Z\boldsymbol{\theta_z}}{(1+\rho)},$$
$$\hat{\boldsymbol{\theta}}_{\boldsymbol{z}} = \text{Lasso}(Z, \boldsymbol{y}_{\boldsymbol{z}}^*, \lambda_z), \text{where} \boldsymbol{y}_{\boldsymbol{z}}^* = \frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)X\boldsymbol{\theta_x}}{(1+\rho)}. \quad [10]$$

This is analogous to the general iterative procedure in Eq. **2**. It is summarized in Algorithm 2.

Algorithm 2 *One-at-a-Time Algorithm for Cooperative Regularized Regression:*

Input: $X \in \mathcal{R}^{n \times p_x}$ and $Z \in \mathcal{R}^{n \times p_z}$, the response $\boldsymbol{y} \in \mathcal{R}^n$, and a grid of hyperparameter values $(\rho_{\texttt{min}}, \ldots, \rho_{\texttt{max}})$.

Fix the lasso penalty weights λ_x and λ_z, **for** $\rho \leftarrow \rho_{\texttt{min}}, \ldots, \rho_{\texttt{max}}$ **do**

Initialize $\boldsymbol{\theta}_{\boldsymbol{x}}^{(0)} \in \mathcal{R}^{p_x}$ and $\boldsymbol{\theta}_{\boldsymbol{z}}^{(0)} \in \mathcal{R}^{p_z}$. **for** $k \leftarrow 0, 1, 2, \ldots$ *until convergence* **do**

Set $\boldsymbol{y}_{\boldsymbol{x}}^* = \frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)Z\boldsymbol{\theta}_{\boldsymbol{z}}^{(k)}}{(1+\rho)}$. Solve Lasso$(X, \boldsymbol{y}_{\boldsymbol{x}}^*, \lambda_x)$ and update $\boldsymbol{\theta}_{\boldsymbol{x}}^{(k+1)}$ to be the solution.

Set $\boldsymbol{y}_{\boldsymbol{z}}^* = \frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)X\boldsymbol{\theta}_{\boldsymbol{x}}^{(k+1)}}{(1+\rho)}$. Solve Lasso$(Z, \boldsymbol{y}_{\boldsymbol{z}}^*, \lambda_z)$ and update $\boldsymbol{\theta}_{\boldsymbol{z}}^{(k+1)}$ to be the solution.

end

end

Select the optimal value of ρ^* based on the sum of the CV errors and get the final fit.

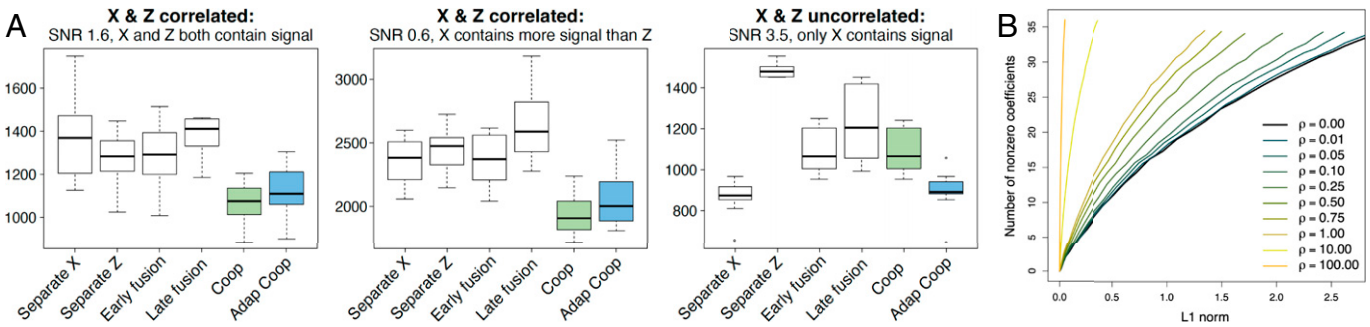

Fig. 2. An illustrative simulation study of cooperative learning in the regression setting and sparsity of the solution. *(A)* Cooperative learning achieves superior prediction accuracy on a test set when the data views *X* and *Z* are correlated. The *y* axis shows the MSE on a test set. The methods in comparison from left to right in each panel correspond to 1) Separate *X*: lasso applied on the data view *X* only; 2) Separate *Z*: lasso applied on the data view *Z* only; 3) Early fusion: lasso applied on the concatenated data views of *X* and *Z*; 4) Late fusion: separate lasso models are fit on *X* and *Z* independently and the predictors are then combined through linear LS; 5) Coop: cooperative learning as outlined in Algorithm 1; and 6) Adap Coop: adaptive cooperative learning, as outlined in Algorithm S2 (*SI Appendix*, section 1). Note that the test MSE in each panel is of a different scale because we experiment with simulating the data of different SNRs. We conducted each simulation experiment 10 times. (*B*) The number of nonzero coefficients as a function of the ℓ_1 norm of the solution with different values of the weight on the agreement penalty term ρ: The solution becomes less sparse as ρ increases.

By iterating back and forth between the two lasso problems, we can find the optimal solution to Eq. **4**. When both X and Z have full column rank, Eq. **4** is strictly convex, and each iteration decreases the overall objective value. Therefore, the one-at-a-time procedure is guaranteed to converge. In general, it can be shown to converge to some stationary point, using results such as those in (21). This algorithm uses fixed values for λ_x, λ_z: we need to run the algorithm over a grid of such values, or use CV to choose λ_x, λ_z within each iteration.

With just two views, there seems to be no advantage to this approach over the direct solution given in Algorithm 1. However, for a larger number of views, there can be a computational advantage, which we will discuss in *Materials and Methods*.

Relation to Early/Late Fusion. From the objective functions Eqs. **3** and **4**, when the weight on the agreement term ρ is set to zero, cooperative learning (regression) reduces to a form of early fusion: We simply concatenate the columns of different views and apply lasso or another regularized regression method.

Next, we discuss the relation of cooperative learning to late fusion. Let X and Z have centered columns and y centered; from Eq. **6**, we obtain

$$\tilde{X}^T\tilde{X} = \begin{pmatrix} X^TX(\mathbf{1}+\rho) & X^TZ(\mathbf{1}-\rho) \\ Z^TX(\mathbf{1}-\rho) & Z^TZ(\mathbf{1}+\rho) \end{pmatrix}. \quad [11]$$

Assuming X and Z have full rank, and omitting the ℓ_1 penalties, we obtain the least-squares estimates

$$\begin{pmatrix} \hat{\boldsymbol{\theta}}_{\boldsymbol{x}} \\ \hat{\boldsymbol{\theta}}_{\boldsymbol{z}} \end{pmatrix} = \begin{pmatrix} X^TX(1+\rho) & X^TZ(1-\rho) \\ Z^TX(1-\rho) & Z^TZ(1+\rho) \end{pmatrix}^{-1} \begin{pmatrix} X^T\boldsymbol{y} \\ Z^T\boldsymbol{y} \end{pmatrix}. \quad [12]$$

If $X^TZ = 0$ (uncorrelated features between the views), this reduces to a linear combination of the least squares estimates for each block; when $\rho = 1$, it is simply the average of the least squares estimates for each block. The above relation also holds when we include the ℓ_1 penalties.

This calculation suggests that restricting ρ to be in $[0, 1]$ would be natural. However, we have found that values larger than one can sometimes yield lower prediction error (*Results*).

Sparsity of the Solution. We explore how the sparsity of the solution depends on the agreement hyperparameter ρ in Fig. 2*B*. We did 100 simulations of Gaussian data with $n = 100$ and $p = 20$ in each of two views, with all coefficients equal to 2.0. The SD of the errors was chosen so that the signal-to-noise ratio (SNR) was about two. The figure shows the number of nonzero coefficients as a function of the overall ℓ_1 of the solutions, for different values of ρ. Note that the lasso parameter λ is varying along the horizontal axis; we chose to plot against the ℓ_1 norm, a more meaningful quantity. We see that the solutions become less sparse as ρ increases, much like the behavior that one sees in the elastic net.

Theoretical Analysis under the Latent Factor Model. To understand the role of the agreement penalty from a theoretical perspective, we consider the following latent factor model. Let $\boldsymbol{u} = (U_1, U_2, \ldots, U_n)$ be a vector of n independent and identically distributed (i.i.d.) random variables with $U_i \sim \mathcal{N}(0, 1)$, $\boldsymbol{y} = (y_1, \ldots, y_n)$, $\boldsymbol{x} = (X_1, \ldots, X_n)$, and $\boldsymbol{z} = (Z_1, \ldots, Z_n)$, with $y_i = \gamma_y U_i + \varepsilon_{yi}$, $X_i = \gamma_x U_i + \varepsilon_{xi}$, and $Z_i = \gamma_z U_i + \varepsilon_{zi}$, where $\varepsilon_{yi} \sim \mathcal{N}\left(0, \sigma_y^2\right)$, $\varepsilon_{xi} \sim \mathcal{N}\left(0, \sigma_x^2\right)$, and $\varepsilon_{zi} \sim \mathcal{N}\left(0, \sigma_z^2\right)$ independently. We show that the mean squared error (MSE) of the predictions from cooperative learning is a decreasing function of ρ around zero with high probability (see details in *SI Appendix*, section 4). Therefore, the agreement penalty offers an advantage in reducing MSE of the predictions under the latent factor model.

Results

Simulation Studies on Cooperative Regularized Linear Regression. Here, we compare cooperative learning in the regression setting with early and late fusion methods in simulations. We generated Gaussian data with $n = 200$ and $p = 500$ in each of two views X and Z and created correlation between them using latent factors. The response $\boldsymbol{y}$ was generated as a linear combination of the latent factors, corrupted by Gaussian noise. We introduced sparsity by letting some columns of X and Z have no effect on $\boldsymbol{y}$. The detailed simulation procedure is outlined in *Materials and Methods*. Datasets are simulated with different levels of correlation between the two data views X and Z, different contributions of X and Z to the signal, and different SNRs. We consider the settings of both small-p and large-p regimes and of both low- and high-SNR ratios. We use 10-fold CV to select the optimal values of hyperparameters.

We compare the following methods: 1) separate X and separate Z: the standard lasso is applied on the separate data views of X and Z with 10-fold CV; 2) early fusion: the standard lasso is applied on the concatenated data views of X and Z with 10-fold CV (note that this is equivalent to cooperative learning with $\rho = 0$); 3) late fusion: separate lasso models are first fitted on X and Z independently with 10-fold CV, and the two resulting predictors

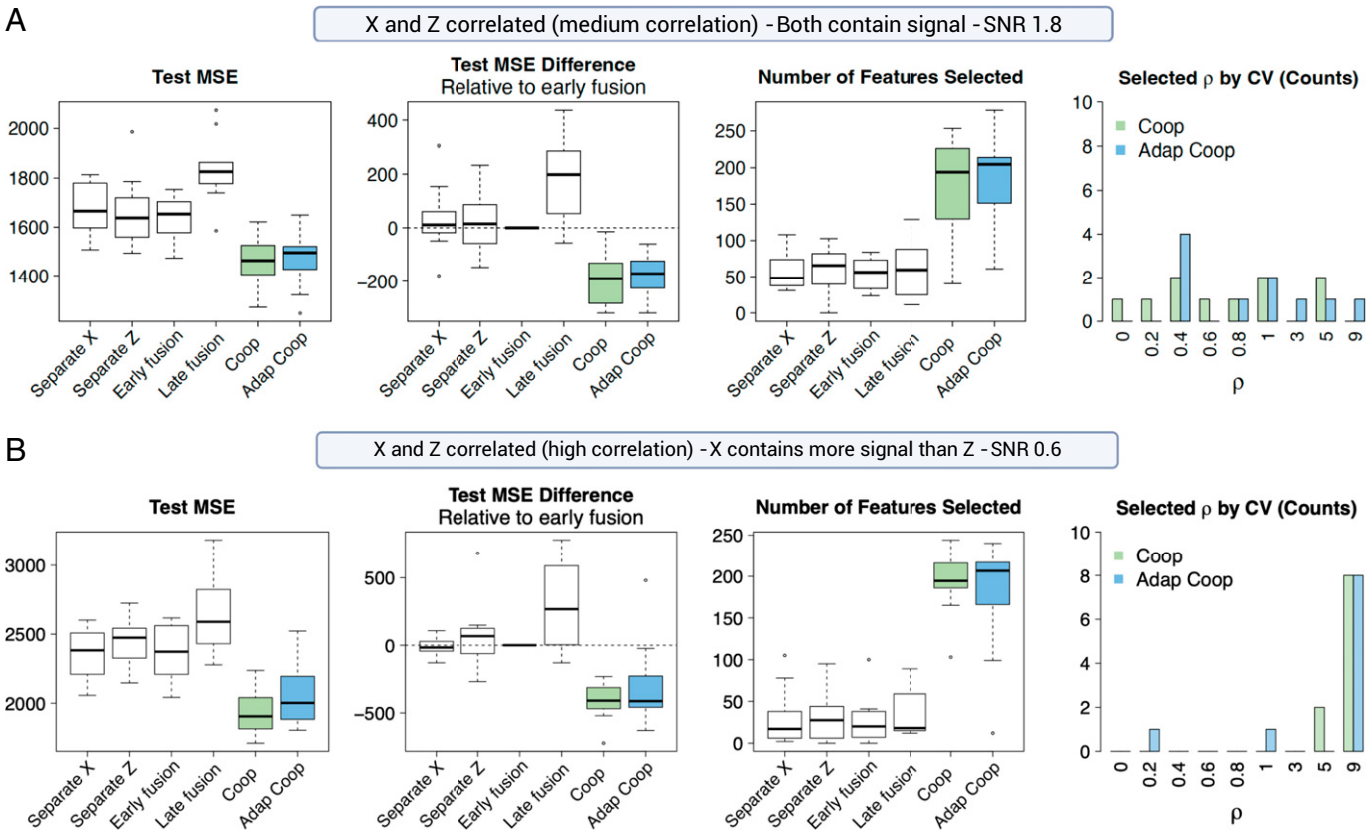

Fig. 3. Simulation studies on cooperative regularized linear regression. (*A*) Simulation results when X and Z have a medium level of correlation and both contain signal ($t_x = t_z = 2$), $n = 200$, $p = 1,000$, and SNR $= 1.8$. The first panel shows MSE on a test set; the second panel shows the MSE difference on the test set relative to early fusion; the third panel shows the number of features selected; and the fourth panel shows the ρ values selected by CV in cooperative learning. Here, "Coop" refers to cooperative learning outlined in Algorithm 1, and "Adap Coop" refers to adaptive cooperative learning outlined in Algorithm S2 (*SI Appendix*, section 1). (*B*) Simulation results when X and Z have a high level of correlation and X contains more signal than Z ($t_x = 6$, $t_z = 1$), $n = 200$, $p = 1,000$, and SNR $= 0.6$.

are then combined through linear least squares (LS) for the final prediction; and 4) cooperative learning (regression) and adaptive cooperative learning. We evaluated the performance based on the MSE on a test set and conducted each simulation experiment 10 times.

Overall, the simulation results can be summarized as follows:

- Cooperative learning performs the best in terms of test MSE across the range of SNR and correlation settings. It is most helpful when the data views are correlated and both contain signal (as in Fig. 3 *A* and *B*). When the correlation between data views is higher, higher values of ρ are more likely to be selected.
- When only one view contains signal and the views are not correlated (*SI Appendix*, Fig. S3*C*), cooperative learning is outperformed by the separate model fit on the view containing the signal, but adaptive cooperative learning is able to perform on par with the separate model, outperforming early and late fusion.
- Moreover, we also find that cooperative learning tends to yield a less sparse model, as expected from the results of *Sparsity of the Solution*.

We include more comprehensive results across a wider range of simulation settings in *SI Appendix*, Figs. S1–S6.

Simulation Studies on Cooperative Learning with Imaging and "Omics" Data. Here, we extend the simulation studies for cooperative learning to the setting where we have two data views of more distinct data modalities, such as imaging and omics data (e.g., transcriptomics and proteomics). We tailor the fitter suitable to each view, i.e., CNNs for images and lasso for omics. We simulate the omics data (X) and the "imaging" data (Z) such that they share some common factors. These factors are also used to generate the signal in the response $\boldsymbol{y}$. We use a factor model to generate the data, as it is a natural way to create correlations between X, Z, and $\boldsymbol{y}$. In *SI Appendix*, section 6, we outline the full details of the simulation procedure. Fig. 4 shows some examples of the synthetic images generated for this study.

Our task is to use the omics and imaging data to predict if a patient has a certain disease. We use a CNN for modeling the imaging data and lasso for the omics data and optimize the

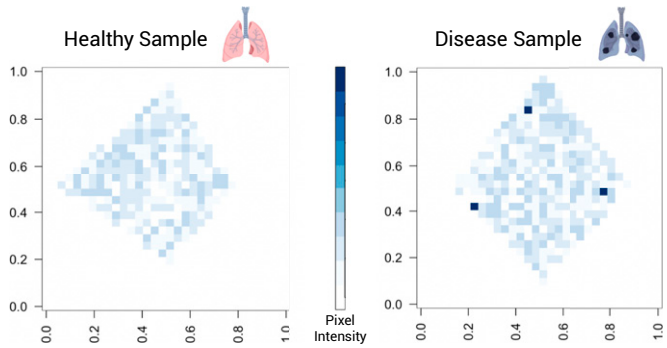

Fig. 4. Generated images for "healthy" and "disease" samples. One can think of the image as an abstract form of a patient's lung, with the darker spots corresponding to the tumor sites. The intensity of the dark spots on the disease samples is generated to correlate with the omics data and the signal in the outcome.

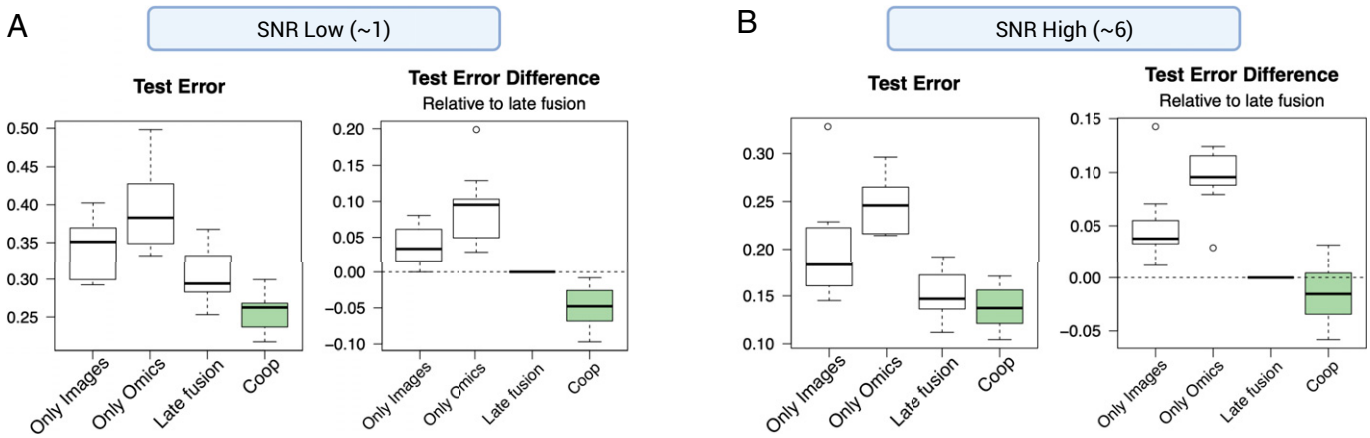

Fig. 5. Simulation studies on cooperative learning with imaging and omics data. *A* corresponds to the relatively low SNR setting (SNR = 1) and *B* to the higher SNR setting (SNR = 6). For each setting, the left panel shows the misclassification error on the test set for CNN on only images, lasso on only omics, late fusion, and cooperative learning; the right panel shows the difference in misclassification error relative to late fusion. Here, Coop refers to cooperative learning. For both settings, the range of ρ values for cooperative learning to select from is (0, 20). The average ρ selected in the low SNR setting is 6.8 and in the high SNR setting is 8.0.

objective for the general form of cooperative learning as in Eq. **1** with the iterative one-at-a-time algorithm outlined in Eq. **2**.

We compare cooperative learning to the following methods: 1) only images: a simple one-layer CNN with max pooling and rectified linear unit activation is applied on the imaging data only; 2) only omics: the standard lasso is applied on the omics data only; and 3) late fusion: separate models (CNN and lasso) are first fit on the imaging and omics data, respectively, and the resulting predictors are then combined through linear LS using a validation set. We evaluated the performance based on the misclassification error on a test set, as well as the difference in misclassification error relative to late fusion.* We consider both low- and high-SNR settings.† We conducted each simulation experiment 10 times.

The results are shown in Fig. 5. We find that 1) late fusion achieves a lower misclassification error on the test set than the separate models; 2) cooperative learning outperforms late fusion and achieves the lowest test error by encouraging the predictions from the two views to agree; and 3) cooperative learning is especially helpful when the SNR is low, while its benefit is less pronounced when the SNR is higher. The last observation makes sense, because when the SNR is lower, the marginal benefit of leveraging the other view(s) in strengthening signal becomes larger.

Multiomics Studies on Labor-Onset Prediction. We applied cooperative learning (regression) to a dataset of labor onset, collected from a cohort of women who went into labor spontaneously, as described in (22). Proteome and metabolome were measured from blood samples collected from the patients during the last 120 d of pregnancy. The goal of the analysis is to predict time to spontaneous labor using proteomics and metabolomics data.

The proteomics data contained measurements for 1,322 proteins, and the metabolomics data contained measurements for 3,529 metabolites. We split the dataset of 53 patients into training and test sets of 40 and 13 patients, respectively.‡ Both the proteomics and metabolomics measurements were screened by their variance across the subjects. We extracted the first time point for each patient from the longitudinal study and predicted the corresponding time to labor. We conducted the same set of experiments across 10 different random splits of the training and test sets.

The results are shown in Table 1. The model fit on the metabolomics data achieves lower test MSE than the one fit on the proteomics data. Early and late fusion hurt performance, as compared to the model fit on only metabolomics. Cooperative learning gives performance gains over the model fit only on metabolomics, outperforming both early and late fusion and achieving the lowest MSE on the test set.

We examined the selected features from cooperative learning and the other methods by comparing the ranking of the features based on the magnitude of their coefficients. All methods rank sialic acid binding immunoglobulin-like lectin-6 (Siglec-6), a protein highly expressed by the placenta (23), as the most important feature for predicting labor onset. As compared to the other methods, cooperative learning boosts up the ranking of features such as plexin-B2 (PLXNB2), which is a protein expressed by the fetal membranes (24), and Activin-A, which is highly expressed by the placenta as well (22). While factors such as Siglec-6, PLXNB2, and Activin-A have previously also been discovered by ref. 22 for labor-onset prediction, C1q was only identified by cooperative learning as 1 of the top 10 features. C1q is an important factor involved in the complement cascade, which influences implantation and fetal development (25), and is worth further investigation for its role in predicting labor onset.

Cooperative Generalized Linear Models and Cox Regression. We next describe how cooperative learning can be extended to generalized linear models (GLMs) (26) and Cox proportional hazards models (27).

Consider a GLM consisting of three components: 1) a linear predictor: $\eta = X\boldsymbol{\beta}$; 2) a link function g such that $\mathrm{E}(Y|X) = g^{-1}(\eta)$; and 3) a variance function as a function of the mean: $V = V(\mathrm{E}(Y|X))$. For cooperative GLMs, we have the linear predictor as $\eta = X\boldsymbol{\theta_x} + Z\boldsymbol{\theta_z}$ and an additional agreement penalty term $\rho||(X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z})||^2$ with the following objective to be minimized:

$$J(\boldsymbol{\theta_x}, \boldsymbol{\theta_z}) = \ell(X\boldsymbol{\theta_x} + Z\boldsymbol{\theta_z}, \boldsymbol{y}) + \frac{\rho}{2}||(X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z})||^2 + \lambda_x||\boldsymbol{\theta_x}||_1 + \lambda_z||\boldsymbol{\theta_z}||_1, \quad [13]$$

*Early fusion is not applicable in this setting.

†The SNR is calculated based on the logits of the probabilities used to generate the class labels.

‡The cohort consisted of 63 patients, as described in (22), but in the public dataset, we only found 53 patients with matched proteomics and metabolomics data.

Table 1. Multiomics studies on labor-onset prediction

Methods	Test MSE		Relative to early fusion		Number of features selected
	Mean	SD	Mean	SD	Mean
Separate proteomics	475.51	80.89	69.14	81.44	26
Separate metabolomics	381.13	36.88	−25.24	30.91	11
Early fusion	406.37	44.77	0	0	15
Late fusion	493.34	63.44	86.97	68.13	21
Cooperative learning	**335.84**	**38.51**	**−70.53**	**32.60**	52

The first two columns in the table show the mean and SD of MSE on the test set across different splits of the training and test sets; the third and fourth columns show the MSE difference relative to early fusion; the last column shows the average number of features selected. The methods include 1) separate proteomics: the standard lasso is applied on the proteomics data only; 2) separate metabolomics: the standard lasso is applied on the metabolomics data only; 3) early fusion: the standard lasso is applied on the concatenated data of proteomics and metabolomics data; 4) late fusion: separate lasso models are first fit on proteomics and metabolomics independently and the predictors are then combined through linear LS; and 5) cooperative learning (Algorithm 1). The average of the selected ρ values is 0.9 for cooperative learning. Cooperative learning achieves the lowest test MSE.

where ℓ is the negative log likelihood of the data. For Cox proportional hazards models, ℓ becomes the negative log partial likelihood of the data.

We make the usual quadratic approximation to Eq. **13**, reducing the minimization problem to a weighted least squares (WLS) problem, which yields

$$\min \frac{1}{2}[||W(\boldsymbol{z} - X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z})||^2 + \rho||(X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z})||^2] + \lambda_x||\boldsymbol{\theta_x}||_1 + \lambda_z||\boldsymbol{\theta_z}||_1, \quad \textbf{[14]}$$

where $\boldsymbol{z}$ is the adjusted dependent variable and W is the diagonal weight matrix, both of which are functions of $\boldsymbol{\theta_x}$ and $\boldsymbol{\theta_z}$.

This leads to an iteratively reweighted least squares (IRLS) algorithm:

- Outer loop: Update the quadratic approximation using the current parameter $\hat{\boldsymbol{\theta}}_x$ and $\hat{\boldsymbol{\theta}}_z$, i.e., update the working response $\boldsymbol{z}$ and the weight matrix W.
- Inner loop: Letting

$$\tilde{X} = \begin{pmatrix} W^{1/2}X & W^{1/2}Z \\ -\sqrt{\rho}X & \sqrt{\rho}Z \end{pmatrix}, \tilde{\boldsymbol{z}} = \begin{pmatrix} W^{1/2}\boldsymbol{z} \\ \boldsymbol{0} \end{pmatrix}, \tilde{\boldsymbol{\beta}} = \begin{pmatrix} \boldsymbol{\theta_x} \\ \boldsymbol{\theta_z} \end{pmatrix}, \quad \textbf{[15]}$$

solve the following problem

$$J(\boldsymbol{\theta_x}, \boldsymbol{\theta_z}) = \frac{1}{2}||\tilde{\boldsymbol{z}} - \tilde{X}\tilde{\boldsymbol{\beta}}||^2 + \lambda_x||\boldsymbol{\theta_x}||_1 + \lambda_z||\boldsymbol{\theta_z}||_1, \quad \textbf{[16]}$$

which is equivalent to Eq. **14**.

Some Extensions

Paired Features from Different Views. One can extend cooperative learning to the setting where a feature in one view is naturally paired with a feature in another view. For example, if the jth column X_j of X is the gene expression for gene j, and Z_k is the expression of the protein k for which gene j codes. In that setup, we would like to encourage agreement between $X_j\boldsymbol{\theta}_{xj}$ and $Z_k\boldsymbol{\theta}_{zk}$. This pairing need not exist for all features, but can occur for a subset of features.

Looking back at our objective function Eq. **4** for two views in the linear case, we add to this objective a pairwise agreement penalty of the form

$$\rho_2 \sum_{j,k \in P} (X_j\boldsymbol{\theta}_{xj} - Z_k\boldsymbol{\theta}_{zk})^2 \quad \textbf{[17]}$$

where P is the set of indices of the paired features.

This additional penalty can be handled easily in the optimization framework. For the direct algorithm (Algorithm 1), we simply add a new row to $\tilde{X}$ and $\tilde{\boldsymbol{y}}$ for each pairwise constraint, while the one-at-a-time algorithm (Algorithm 2) can be similarly modified.

Modeling Interactions between Views. In our general objective function Eq. **1**, we can capture interactions between features in the same view, by using methods such as random forests or boosting for the learners f_X and f_Z. However, this framework does not allow for interactions between features in different views. Here is an objective function to facilitate such interactions:

$$\min \mathrm{E}\Big[\frac{1}{2}(\boldsymbol{y} - f_X(X) - f_Z(Z) - f_{XZ}(X,Z))^2 + \frac{\rho}{2}(f_X(X) - f_Z(Z))^2 + \frac{\rho}{2(1-\rho)}f_{XZ}^2(X,Z)\Big], \quad \textbf{[18]}$$

where $f_{XZ}(X,Z)$ is a joint function of X and Z, including, for example, interactions between the features in each view.

The solution to Eq. **18** has fixed points:

$$f_X(X) = \mathrm{E}\Big[\frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)f_Z(Z)}{(1+\rho)} - \frac{f_{XZ}(X,Z)}{1+\rho}|X\Big],$$
$$f_Z(Z) = \mathrm{E}\Big[\frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)f_X(X)}{(1+\rho)} - \frac{f_{XZ}(X,Z)}{1+\rho}|Z\Big],$$
$$f_{XZ}(X,Z) = \mathrm{E}\Big[(1-\rho)(\boldsymbol{y} - f_X(X) - f_Z(Z))|X,Z\Big]. \quad \textbf{[19]}$$

When $\rho = 0$, from Eq. **18**, the solution reduces to the additive model $f_X(X) + f_Z(Z) + f_{XZ}(X,Z)$. As $\rho \to 1$, the joint term $f_{XY} \to 0$, and we again get the late fusion estimate as the average of the marginal predictions $\hat{f}_X(X)$ and $\hat{f}_Z(Z)$. To implement this in practice, we simply insert learners such as random forest or boosting for f_X, f_Z and f_{XZ}. The first two use only features from X and Z, while the last uses features from both.

Discussion

In this paper, we introduce a method called cooperative learning for supervised learning with multiple sets of features, or "data views." The method encourages the predictions from different data views to align through an agreement penalty. By varying the weight of the agreement penalty in the objective, we obtain a spectrum of solutions that include the commonly used early and late fusion methods. The method can choose the degree of agreement (or fusion) in an data-adaptive manner.

Cooperative learning provides a powerful tool for multiomics data fusion by strengthening aligned signals across modalities and

allowing flexible fitting mechanisms for different modalities. The effectiveness of our methodology has implications for improving diagnostics and therapeutics in an increasingly multiomic world.

Furthermore, cooperative learning could be extended to the semisupervised setting when we have additional matched data views on unlabeled samples. The agreement penalty allows us to leverage the signals in the matched unlabeled samples to our advantage. In addition, when we have missing values in some data views, the agreement penalty also allows us to impute one view from the other(s). Lastly, the method can be easily extended to binary, count, and survival data.

Materials and Methods

Cooperative Learning with More Than Two Data Views. When we have more than two views of the data, $X_1 \in \mathcal{R}^{n\times p_1}, X_2 \in \mathcal{R}^{n\times p_2}, \ldots, X_M \in \mathcal{R}^{n\times p_M}$, the population quantity that we want to minimize becomes

$$\min \mathrm{E}\Big[\frac{1}{2}(\boldsymbol{y} - \sum_{m=1}^{M} f_{X_m}(X_m))^2 + \frac{\rho}{2}\sum_{m<m'}(f_{X_m}(X_m) - f_{X_{m'}}(X_{m'}))^2\Big]. \quad \textbf{[20]}$$

We can also have different weights on the agreement penalties for distinct pairs of data views, forcing some pairs to agree more than others. In addition, we can incorporate prior knowledge in determining the relative strength of the agreement penalty for each pair of views.

As with two views, this can be optimized with an iterative algorithm that updates each $f_{X_m}(X_m)$ as follows:

$$f_{X_m}(X_m) = \mathrm{E}\Big[\frac{\boldsymbol{y}}{1+(M-1)\rho} - \frac{(1-\rho)\sum_{m'\neq m} f_{X_{m'}}(X_{m'})}{1+(M-1)\rho}|X_m\Big]. \quad \textbf{[21]}$$

As in the two-view setup above, the fitter $E(\cdot|X_m)$ can be tailored to the data type of each view.

For regularized linear regression with more than two views, the objective becomes

$$J(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_M) = \frac{1}{2}||\boldsymbol{y} - \sum_{m=1}^{M} X_m\boldsymbol{\theta}_m||^2 + \frac{\rho}{2}\sum_{m<m'}||(X_m\boldsymbol{\theta}_m - X_{m'}\boldsymbol{\theta}_{m'})||^2 + \sum_{m=1}^{M}\lambda_m||\boldsymbol{\theta}_m||_1. \quad \textbf{[22]}$$

This is, again, a convex problem. The optimal solution can be found by forming augmented data matrices as before in Eqs. **6** and **7**.

Let

$$\tilde{X} = \begin{pmatrix} X_1 & X_2 & \ldots & X_{M-1} & X_M \\ -\sqrt{\rho}X_1 & \sqrt{\rho}X_2 & \ldots & 0 & 0 \\ -\sqrt{\rho}X_1 & 0 & \ldots & \sqrt{\rho}X_{M-1} & 0 \\ -\sqrt{\rho}X_1 & 0 & \ldots & 0 & \sqrt{\rho}X_M \\ 0 & -\sqrt{\rho}X_2 & \ldots & \sqrt{\rho}X_{M-1} & 0 \\ 0 & -\sqrt{\rho}X_2 & \ldots & 0 & \sqrt{\rho}X_M \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & -\sqrt{\rho}X_{M-1} & \sqrt{\rho}X_M \end{pmatrix},$$

$$\tilde{\boldsymbol{y}} = (\boldsymbol{y} \quad \mathbf{0} \quad \ldots \quad \mathbf{0})^T, \ \tilde{\boldsymbol{\beta}} = (\boldsymbol{\theta}_1 \quad \boldsymbol{\theta}_2 \quad \ldots \quad \boldsymbol{\theta}_M)^T, \quad \textbf{[23]}$$

then the equivalent problem to Eq. **22** becomes

$$\frac{1}{2}||\tilde{\boldsymbol{y}} - \tilde{X}\tilde{\boldsymbol{\beta}}||^2 + \sum_{m=1}^{M}\lambda_m||\boldsymbol{\theta}_m||_1. \quad \textbf{[24]}$$

With M views, the augmented matrix in Eq. **23** has $n + \binom{M}{2}\cdot n$ rows, which could be computationally challenging to solve.

Alternatively, the optimal solution $\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2, \ldots, \hat{\boldsymbol{\theta}}_M$ has fixed points

$$\hat{\boldsymbol{\theta}}_m = \mathrm{Lasso}(X, \boldsymbol{y}_m^*, \lambda_m), \quad \text{where } \boldsymbol{y}_m^* = \frac{\boldsymbol{y}}{1+(M-1)\rho} - \frac{(1-\rho)\sum_{m'\neq m} X_{m'}\boldsymbol{\theta}_{m'}}{1+(M-1)\rho}. \quad \textbf{[25]}$$

This leads to an iterative algorithm, where we successively solve each subproblem until convergence. For a large number of views, this can be a more efficient procedure than the direct approach in Eq. **24**. We include simulation studies on cooperative learning for more than two views in *SI Appendix*, section 3.

Simulation Procedure for Cooperative Regularized Linear Regression. The simulation is set up as follows. Given values for parameters $n, p_x, p_z, p_u, s_u, t_x, t_z, \boldsymbol{\beta}_u, \sigma$, we generate data according to the following procedure:

1. $x_j \in \mathcal{R}^n$ distributed i.i.d. MVN$(0, I_n)$ for $j = 1, 2, \ldots, p_x$.
2. $z_j \in \mathcal{R}^n$ distributed i.i.d. MVN$(0, I_n)$ for $j = 1, 2, \ldots, p_z$.
3. For $i = 1, 2, \ldots, p_u$ (p_u corresponds to the number of latent factors, $p_u < p_x$ and $p_u < p_z$):
   a) $u_i \in \mathcal{R}^n$ distributed i.i.d. MVN$(0, s_u^2 I_n)$;
   b) $x_i = x_i + t_x * u_i$;
   c) $z_i = z_i + t_z * u_i$.
4. $X = [x_1, x_2, \ldots, x_{p_x}], Z = [z_1, z_2, \ldots, z_{p_z}]$.
5. $U = [u_1, u_2, \ldots, u_{p_u}]$, $\boldsymbol{y} = U\boldsymbol{\beta}_u + \epsilon$ where $\epsilon \in \mathcal{R}^n$ distributed i.i.d. MVN$(0, \sigma^2 I_n)$.

There is sparsity in the solution since a subset of columns of X and Z are independent of the latent factors used to generate $\boldsymbol{y}$.

Relation to existing approaches. We have mentioned the close connection of cooperative learning to early and late fusion: Setting $\rho = 0$ or 1 gives a version of each of these, respectively. There are many variations of late fusion, including the use of stacked generalization to combine the predictions at the last stage (28).

Cooperative learning is also related to "collaborative regression" (29). This method uses an objective function of the form

$$\frac{b_{xy}}{2}||\boldsymbol{y} - X\boldsymbol{\theta_x}||^2 + \frac{b_{zy}}{2}||\boldsymbol{y} - Z\boldsymbol{\theta_z}||^2 + \frac{b_{xz}}{2}||X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z}||^2. \quad \textbf{[26]}$$

With ℓ_1 penalties added, this is proposed as a method for sparse supervised canonical correlation analysis. It is different from cooperative learning in an important way: Here, X and Z are not fit jointly to the target. The authors state that collaborative regression is not well suited to the prediction task. We note that if $b_{xy} = b_{zy} = b_{xz} = 1$, each of $\hat{\boldsymbol{\theta}}_{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}_{\boldsymbol{z}}$ are one-half of the LS estimates on X, Z, respectively. Hence, the overall prediction $\hat{\boldsymbol{y}}$ is the average of the individual LS predictions. This late fusion estimate is the same as that obtained from cooperative learning with $\rho = 1$. In addition, a related framework based on optimizing measures of agreement between data views was also proposed in (30), but it is different from cooperative learning in the sense that the data views are not used jointly to model the target.

Cooperative learning also has connections with "contrastive learning" (18, 19). This method is an unsupervised learning technique first proposed for learning visual representations. Without the supervision of $\boldsymbol{y}$, it learns representations of images by maximizing agreement between differently augmented "views" of the same data example. While both contrastive learning and cooperative learning have a term in the objective that encourages agreement between correlated views, our method combines the agreement term with the usual prediction error loss and is thus supervised.

Moreover, the iteration Eq. **2** looks much like the backfitting algorithm for "generalized additive models" (31). In that setting, each of f_X and f_Z are typically functions of one-dimensional features X and Z, and the backfitting algorithm iterations correspond to Eq. **2** with $\rho = 0$. In the additive model setting, backfitting is a special case of the Gauss–Seidel algorithm (31). In cooperative learning, each of X, Z are views with multiple features; we could use an additive model for each view, i.e., $f_X(X) = \sum_i g_i(X_i)$, $f_Z(Z) = \sum_j h_j(Z_j)$, where i and j are column indices of X and Z, respectively. Then each of the iterations in Eq. **2** could be solved by using a backfitting algorithm, leading to a nested procedure.

We next discuss the relation of cooperative learning to a recently proposed method for multiview analysis called "sparse integrative discriminant analysis" (SIDA) (32). This method aims to identify variables that are associated across views, while also able to optimally separate data points into different classes. Specifically, it combines canonical correlation analysis and linear discriminate analysis by solving the following optimization problem. Let $X_k = (\boldsymbol{x}_{1k}, \ldots, \boldsymbol{x}_{n_k,k})^T \in \mathcal{R}^{n_k \times p}$, $\boldsymbol{x}_k \in \mathcal{R}^p$ be the data matrix for class k, where $k = 1, \ldots, K$, and n_k is the number of samples in class k. Then, the mean vector for class k is $\hat{\mu}_k = \frac{1}{n_k}\sum_{i=1}^{n_k} \boldsymbol{x}_{ik}$; the common variance matrix for all classes is $S_w = \sum_{k=1}^{K}\sum_{i=1}^{n}(\boldsymbol{x}_{ik} - \hat{\mu}_k)(\boldsymbol{x}_{ik} - \hat{\mu}_k)^T$; the between-class covariance matrix is $S_b = \sum_{k=1}^{K} n_k(\hat{\mu}_k - \hat{\mu})(\hat{\mu}_k - \hat{\mu})^T$, where $\hat{\mu} = \frac{1}{n}\sum_{k=1}^{K} n_k\hat{\mu}_k$ is the combined class mean vector. Assume that we have two data views $X \in \mathcal{R}^{n \times p_x}$ and $Z \in \mathcal{R}^{n \times p_z}$ with centered columns, we want to find $A = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{K-1}]$ and $B = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{K-1}]$ such that

$$\max \rho \cdot \mathrm{tr}(A^T S_b^x A + B^T S_b^z B) + (1-\rho) \cdot \mathrm{tr}(A^T S_{xz} B B^T S_{xz}^T A)$$
$$\text{s.t. } \mathrm{tr}(A^T S_w^x A)/(K-1) = 1 \,\&\, \mathrm{tr}(B^T S_w^z B)/(K-1) = 1,$$

where $S_{xz} \in \mathcal{R}^{p_x \times p_z}$ is the sample cross-covariance matrix between X and Z. Here, $\mathrm{tr}(\cdot)$ is the trace function, and ρ is the parameter that controls the relative importance of the "separation" term and the "association" terms in the objective. While SIDA also considers the association across data views by choosing vectors that are associated and able to separate data points into classes, it solves the problem in a "backward" manner–that is, the features are modeled as a function of the outcome. Cooperative learning, in contrast, solves the problem in a "forward" manner ($Y \sim X, Z$), which is more suitable for prediction.

We also note the connection between cooperative learning (regression) with the "standardized group lasso" (33). This method is a variation of the group lasso (34) and uses

$$\|X\boldsymbol{\theta_x}\|_2 + \|Z\boldsymbol{\theta_z}\|_2 \qquad \textbf{[27]}$$

as the penalty term, rather than the sum of squared two norms. It encourages group-level sparsity by eliminating entire blocks of features at a time. In the group lasso, each block is a group of features, and we do not expect each block to be predictive on its own. This is different from cooperative learning, where each feature block is a data view, and we generally do not want to eliminate an entire view for prediction. In addition, the standardized group lasso does not have an agreement penalty. One could, in fact, add the standardized group lasso penalty (35) to the cooperative learning objective, which would allow elimination of entire data views.

Data, Materials, and Software Availability. The data associated with the labor-onset study (22) can be obtained via Zenodo (doi: 10.5281/zenodo.4509768). The code used to perform the study has been deposited onto the cooperative-learning GitHub repository (https://github.com/dingdaisy/cooperative-learning) (36). An open-source R language package for cooperative learning called "multiview" is available on the CRAN repository. (https://cran.r-project.org/web/packages/multiview) (37).

ACKNOWLEDGMENTS. We thank Olivier Gevaert, Trevor Hastie, Ryan Tibshirani, and Samson Mataraso for helpful discussions and two referees whose comments greatly improved this manuscript. D.Y.D was supported by the Stanford Graduate Fellowship. B.N. was supported by Stanford Clinical & Translational Science Award Grant 5UL1TR003142-02 from the NIH National Center for Advancing Translational Sciences. R.T. was supported by NIH Grant 5R01 EB001988-16 and NSF Grant 19 DMS1208164.

1. V. N. Kristensen *et al.*, Principles and methods of integrative genomic analyses in cancer. *Nat. Rev. Cancer* **14**, 299–313 (2014).
2. M. D. Ritchie, E. R. Holzinger, R. Li, S. A. Pendergrass, D. Kim, Methods of integrating data to uncover genotype-phenotype interactions. *Nat. Rev. Genet.* **16**, 85–97 (2015).
3. D. R. Robinson *et al.*, Integrative clinical genomics of metastatic cancer. *Nature* **548**, 297–303 (2017).
4. K. J. Karczewski, M. P. Snyder, Integrative omics for health and disease. *Nat. Rev. Genet.* **19**, 299–310 (2018).
5. A. Ma, A. McDermaid, J. Xu, Y. Chang, Q. Ma, Integrative methods and practical challenges for single-cell multi-omics. *Trends Biotechnol.* **38**, 1007–1022 (2020).
6. Y. Hao *et al.*, Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573–3587.e29 (2021).
7. Y. Yuan *et al.*, Assessing the clinical utility of cancer genomic and proteomic data across tumor types. *Nat. Biotechnol.* **32**, 644–652 (2014).
8. A. J. Gentles *et al.*, Integrating tumor and stromal gene expression signatures with clinical indices for survival stratification of early-stage non–small cell lung cancer. *J. Natl. Cancer Inst.* **107**, djv211 (2015).
9. B. A. Perkins *et al.*, Precision medicine screening using whole-genome sequencing and advanced imaging to identify disease risk in adults. *Proc. Natl. Acad. Sci. U.S.A.* **115**, 3686–3691 (2018).
10. K. Chaudhary, O. B. Poirion, L. Lu, L. X. Garmire, Deep learning–based multi-omics integration robustly predicts survival in liver cancer. *Clin. Cancer Res.* **24**, 1248–1259 (2018).
11. S. Wold, K. Esbensen, P. Geladi, Principal component analysis. *Chemom. Intell. Lab. Syst.* **2**, 37–52 (1987).
12. P. Vincent *et al.*, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **11**, 3371–3408 (2010).
13. P. Yang, Y. Hwa Yang, B. B. Zhou, A. Y. Zomaya, A review of ensemble methods in bioinformatics. *Curr. Bioinform.* **5**, 296–308 (2010).
14. J. Zhao *et al.*, Learning from longitudinal data in electronic health record and genetic data to improve cardiovascular event prediction. *Sci. Rep.* **9**, 717 (2019).
15. R. J. Chen *et al.*, Pathomic fusion: An integrated framework for fusing histopathology and genomic features for cancer diagnosis and prognosis. *IEEE Trans. Med. Imaging* **41**, 757–770 (2022).
16. J. J. Chabon *et al.*, Integrating genomic features for non-invasive early lung cancer detection. *Nature* **580**, 245–251 (2020).
17. L. Wu *et al.*; PRACTICAL consortium; CRUK Consortium; BPC3 Consortium; CAPS Consortium; PEGASUS Consortium, An integrative multi-omics analysis to identify candidate DNA methylation biomarkers related to prostate cancer risk. *Nat. Commun.* **11**, 3905 (2020).
18. T. Chen, S. Kornblith, M. Norouzi, G. Hinton, "A simple framework for contrastive learning of visual representations" in *International Conference on Machine Learning* (PMLR, 2020), pp. 1597–1607.
19. P. Khosla *et al.*, "Supervised contrastive learning" in *Proceedings of the 34th Conference on Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, H. Lin, Eds. (Advances in Neural Information Processing Systems, Curran Associates, Inc., Red Hook, NY, 2020), vol. 33, pp. 18661–18673.
20. J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* **33**, 1–22 (2010).
21. R. J. Tibshirani, Dykstra's algorithm, admm, and coordinate descent: Connections, insights, and extensions. arXiv [Preprint] (2017). https://arxiv.org/abs/1705.04768. Accessed 12 May 2017.
22. I. A. Stelzer *et al.*, Integrated trajectories of the maternal metabolome, proteome, and immunome predict labor onset. *Sci. Transl. Med.* **13**, eabd9898 (2021).
23. E. C. Brinkman-Van der Linden *et al.*, Human-specific expression of Siglec-6 in the placenta. *Glycobiology* **17**, 922–931 (2007).
24. H. Singh, J. D. Aplin, Endometrial apical glycoproteomic analysis reveals roles for cadherin 6, desmoglein-2 and plexin b2 in epithelial integrity. *Mol. Hum. Reprod.* **21**, 81–94 (2015).
25. G. Girardi, J. J. Lingo, S. D. Fleming, J. F. Regal, Essential role of complement in pregnancy: From implantation to parturition and beyond. *Front. Immunol.* **11**, 1681 (2020).
26. J. A. Nelder, R. W. Wedderburn, Generalized linear models. *J. Royal Stat. Soc. Ser. A (General)* **135**, 370–384 (1972).
27. D. R. Cox, Regression models and life-tables. *J. Royal Stat. Soc. Ser. B (Methodological)* **34**, 187–202 (1972).
28. E. Garcia-Ceja, C. E. Galván-Tejada, R. Brena, Multi-view stacking for activity recognition with sound and accelerometer data. *Inf. Fusion* **40**, 45–56 (2018).
29. S. M. Gross, R. Tibshirani, Collaborative regression. *Biostatistics* **16**, 326–338 (2015).
30. V. Sindhwani, P. Niyogi, M. Belkin, "A co-regularization approach to semi-supervised learning with multiple views" in *Proceedings of ICML Workshop on Learning with Multiple Views* (Citeseer, 2005), Vol. 2005, pp. 74–79.
31. T. J. Hastie, R. J. Tibshirani, *Generalized Additive Models* (CRC Press, Boca Raton, FL, 1990).
32. S. E. Safo, E. J. Min, L. Haine, Sparse linear discriminant analysis for multiview structured data. *Biometrics* **78**, 612–623 (2022).
33. N. Simon, R. Tibshirani, Standardization and the group lasso penalty. *Stat. Sin.* **22**, 983–1001 (2012).
34. M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Series B Stat. Methodol.* **68**, 49–67 (2006).
35. M. Ponzetti *et al.*, Non-conventional role of haemoglobin beta in breast malignancy. *Br. J. Cancer* **117**, 994–1006 (2017).
36. D. Y. Ding, Cooperative Learning for Multi-view Analysis. GitHub. https://github.com/dingdaisy/cooperative-learning. Deposited 18 May 2022.
37. D. Y. Ding et al., Data from "Cooperative learning for multiview analysis." CRAN repository. https://cran.r-project.org/web/packages/multiview. Accessed 29 August 2022.