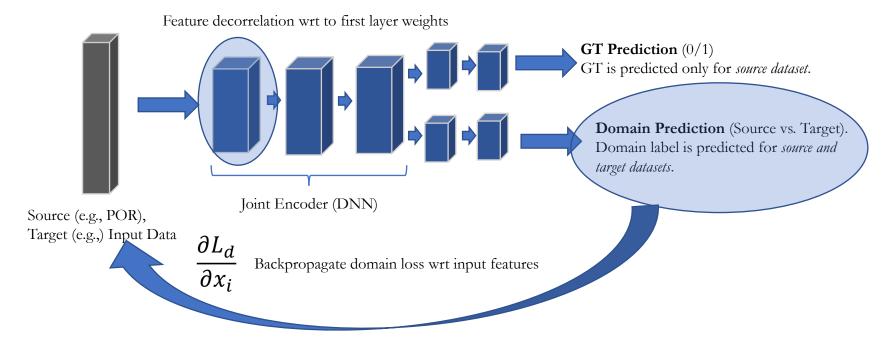
Decorrelated Distribution Shift Feature Ranking (DDSFR)

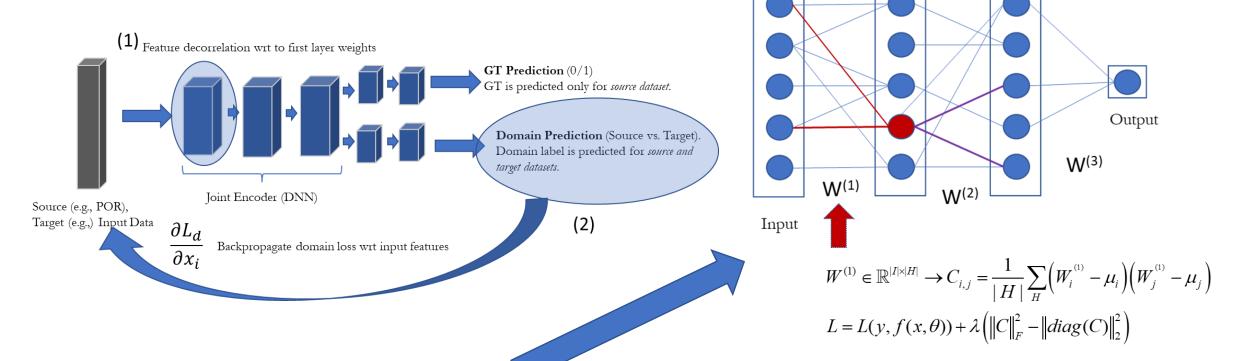


Core idea: We aim to identify the *top-k* most important features for a source \rightarrow target domain shift setting; we assume that we have the GT values for the source dataset, but no GT values are available for the target dataset. This way the model learns to extrapolate GT predictions for the target, shifted data, in addition to predicting the data domain.

We construct a joint encoder model to simultaneously predict the domain (source vs. target) and GT values (only for the source dataset).



Decorrelated Distribution Shift Feature Ranking (DDSFR)

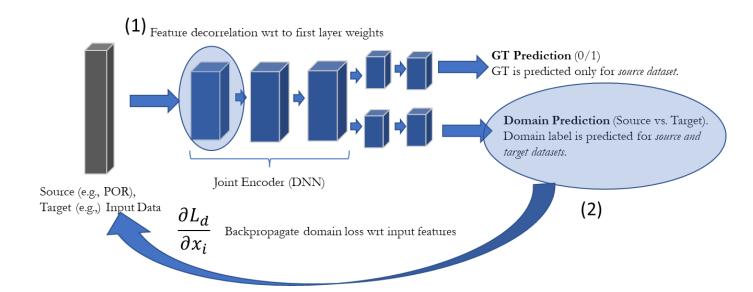


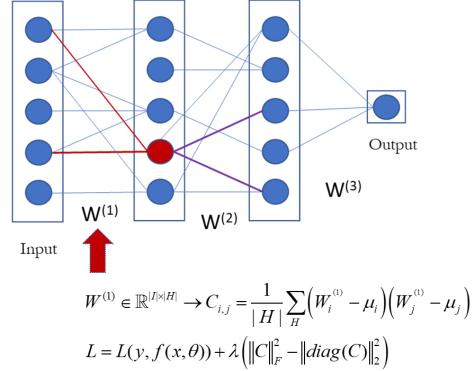
Two ingredients:

(1) Neural Feature Decorrelation: we enforce seamless feature decorrelation via an auxiliary function that minimizes the off-diagonal feature covariances of the first layer weights of the model.



Decorrelated Distribution Shift Feature Ranking (DDSFR)



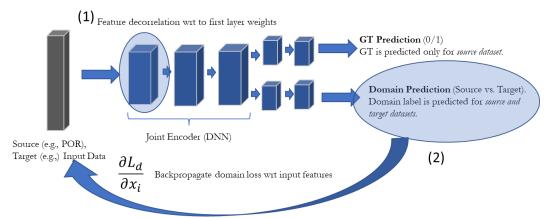


Two ingredients:

- (1) Neural Feature Decorrelation: we enforce seamless feature decorrelation via an auxiliary function that minimizes the off-diagonal feature covariances of the first layer weights of the model.
- **(2) Backpropagation of the domain loss wrt to input features**; gradients are averaged over the entire test dataset to calculate DDSFR scores.



Decorrelated Distribution Shift Feature Ranking (DDSFR)



Data: 80% training / 20% test split; Data-Preprocessing: Cols removed as identified by Mark; NaN GT data pts removed; input to model is full ~3k data features; missing data features imputed with feature mean; GT='GT_SDT'; note that the model is not explicitly trained on GT annotations for the target dataset.

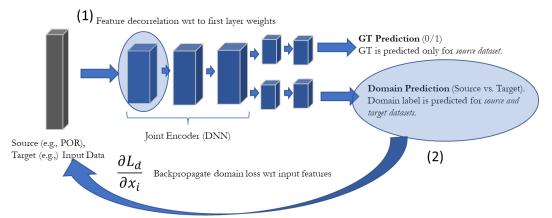
Results: POR (source) vs. HDR fan swap (target)

Test Prediction Task	AUC
GT prediction for <i>source</i> (POR)	99.09
GT prediction for target	96.92
Domain prediction (source vs. target)	97.89



intel

Decorrelated Distribution Shift Feature Ranking(DDSFR)



Data: 80% training / 20% test split; Data-Preprocessing: Cols removed as identified by Mark; NaN GT data pts removed; input to model is full ~3k data features; missing data features imputed with feature mean; GT='GT_SDT'; note that the model is not explicitly trained on GT annotations for the target dataset.

Results: POR (source) vs. HDR fan no anneal (target)

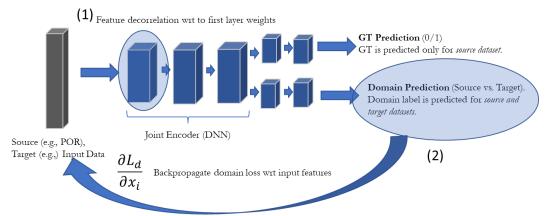
Test Prediction Task	AUC
GT prediction for <i>source</i> (POR)	99.04
GT prediction for target	97.13
Domain prediction (source vs. target)	96.98



Top-500 ranked features for DDSFR (see txt file)



Decorrelated Distribution Shift Feature Ranking(DDSFR)



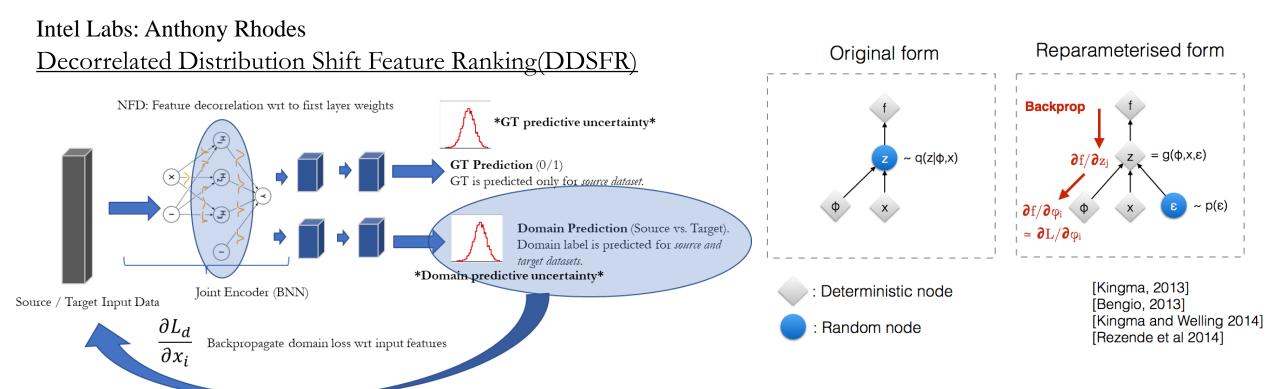
Data: 80% training / 20% test split; Data-Preprocessing: NaN, GT data pts removed; input to model top 100 common data features – as ranked by LightGBM (~300k+ data pts); missing data features imputed with feature mean; GT='GT_Hot'; note that the model is not explicitly trained on GT annotations for the target dataset.

Results: new_report (source) vs. RPL81(target)

Test Prediction Task	AUC
GT prediction for source	71.03
GT prediction for target	80.57
Domain prediction (source	1.0
vs. target)	



Top-100 ranked features for DDSFR (see txt file)



DDSFR: Feature Importance Ranking

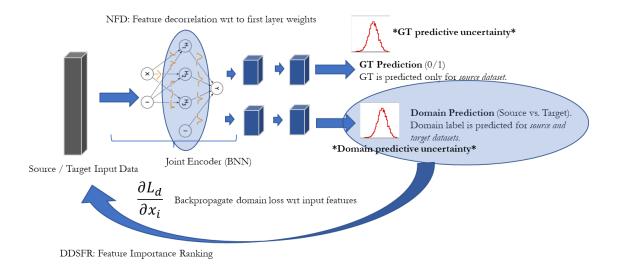
Predictive uncertainty estimation: We can easily adapt our NN-based solution for feature ranking to elicit predictive uncertainty estimates, e.g., using **BNN**s (code already implemented) or **MC Dropout** for predictive uncertainty estimation for GT and domain prediction.

*Such uncertainty estimates can be used to further enhance DDSFR explainability, amplify human-in-the-loop functionality, help with OOD/outlier detection.

Q: For the yield use case, which settings are most useful: uncertainty estimate for domain/GT, per-datum, feature ranking over entire dataset or per-datum, others?



Decorrelated Distribution Shift Feature Ranking(DDSFR)



Overview of pros/cons of this method:

- (+) Single, end-to-end model for multi-task prediction (GT/domain prediction) and feature ranking
- (+) Model learns to reliably extrapolate unsupervised GT prediction for target domain
- (+) Embedded feature decorrelation to reduce redundancy
- (+) Easy to adapt predictive uncertainty estimates
- (-) Requires a bit more compute/train time than some classical models (lightGBM)

