Table 3: Datasets, models and neighborhoods used in experiments. RF→ Random Forest, NN→ Neural Network, ResNet→ Residual Network and NB→ Naive Bayes.

Dataset	Modality	Black-box model acc/R^2,	Realistic neighborhood creation methods
IRIS	tabular	RF classifier, 93%	KDEGen [8], RF [36]
MEPS	tabular	RF regressor, 0.325	[36]
FMNIST	image	NN classifier, 87%	VAEGen [8]
CIFAR10	image	ResNet18, 95%	VAEGen [8]
Rotten Tomatoes	text	NB classifier, 75%	Word2VecGen [8]

A Efficiency of LINEX

It is important to note that the query complexity (i.e. number of times we query the black box to obtain an explanation) of LINEX is the same as that of LIME since the union of the environments is the same as a LIME perturbation neighborhood. This is important in todays cloud-driven world where models may exist on different cloud platforms and posthoc explanations are an independent service where each call to the model has an associated cost. In terms of running time for two environments, convergence was fast and running time was approximately 2.5 times that of LIME (LINEX took 2.5 seconds on IRIS for 30 examples as opposed to 1 second by LIME, LINEX took 47 seconds on MEPS for 500 examples as opposed to 18 seconds by LIME), which is very similar to Smoothed LIME (S-LIME) (took 2.3 seconds on IRIS and 40 seconds on MEPS) that we still outperform in majority of the cases.

Realistic neighborhood generation can be time consuming especially for MeLIME since generators have to be trained which may take up to an hour using a single GPU for datasets such as FMNIST. After the generator is trained and neighborhood sampled MeLIME takes the same amount of time as LIME since the model fitting procedure is the same. MAPLE took 1.5 seconds for the IRIS dataset for 30 examples and 27 seconds for 500 MEPS examples.

A way to further speed up LINEX would be to implement it through *embarrassing parallelism* which can easily be done across explanations. This will prevent scaling of the running time in the number of examples when many explanations are needed. The setting with many explanations is anyway where we would need efficiency because if only few explanations were desired the slightly higher running time of LINEX would not be an issue.

B Proof of Theorem 1

Expanding on the proof sketch provided in the main paper we now provide a case wise analysis to prove Theorem 1.

• $\boldsymbol{w}_1^* = \boldsymbol{w}_2^*$: If the optimal solutions to both environments in the convex set $[-\gamma, \gamma]^d$ are the same, then in the first iteration itself where we fit to the first environment we would have reached the optimal solution to our problem where $\tilde{\boldsymbol{w}}_1 = \boldsymbol{w}_1^*$. This is because in the second iteration where we fit the second environment to the residual from the previous fit $\tilde{\boldsymbol{w}}_2 = \mathbf{0}$ and the algorithm would terminate. This would imply the output of algorithm 1 would be $\boldsymbol{w} = \boldsymbol{w}_1^*$.

• $\boldsymbol{w}_1^* \neq \boldsymbol{w}_2^*$: When the optimal solutions for the two environments are not equal we consider the following two cases:

- **Opposite sign attributions:** If the i^{th} component of $\boldsymbol{w}_1^*$ and $\boldsymbol{w}_2^*$ have opposite signs, then the i^{th} components of the ensemble predictor, $\tilde{w}_{1i}$ and $\tilde{w}_{2i}$ are both at the boundary γ and $-\gamma$ respectively if $\tilde{w}_{1i} > 0$. This is because both try to push the ensemble (i.e. their sum) towards the sign they have where eventually they reach the boundary $\pm\gamma$ and have no incentive to deviate. Any deviation from these values will lead to a higher least squares error in their environment, thus making this a NE.
- **Same sign attributions:** If the i^{th} component of $\boldsymbol{w}_1^*$ and $\boldsymbol{w}_2^*$ have same signs, then the i^{th} component of ensemble predictor constructed from the NE is set to the least squares attribution with a smaller absolute value, i.e., $w_i = w_{1i}^*$, where $|w_{1i}^*| \leq |w_{2i}^*|$. Without loss of generality assume $0 < w_{1i}^* < w_{2i}^*$, the attribution of the environments' predictors in NE, then $\tilde{w}_{1i}$ and $\tilde{w}_{2i}$ have opposite signs, i.e., $\tilde{w}_{2i} = \gamma$ and $\tilde{w}_{1i} = w_{1i}^* - \gamma$ where the

ensemble predictor for the i^{th} component would be $w_i = \tilde{w}_{1i} + \tilde{w}_{2i} = w^*_{1i} - \gamma + \gamma = w^*_{1i}$, since any deviation from this would lead to a worse least squares loss for the corresponding environment. This shows that ensemble predictor is conservative and selects the smaller least squares attribution.

C Behavior for More than Two Environments

Given Assumptions 1 and 2 we now discuss the behavior of our method for more than two environments. If the number of environments is odd, then using similar logic to that discussed in the proof sketch one can see that the feature attribution would be equal to the median of the feature attributions across all the environments. Essentially, all environments with optimal least squares attributions above the median would be at $+\gamma$, while those below it would be at $-\gamma$. The one at the median would remain so with no incentive for any environment to alter its attribution making it a NE. This is a stable choice that is also likely to be faithful as we have no more information to decide otherwise. On the other hand if we have an even number of environments the final attribution in this case depends on the middle two environments in the same manner as the two environment case proved in Theorem 1. Thus, if the optimal least squares attributions of the middle two environments have opposite sign, then the final attribution is zero, else its the lower of the two attributions in terms of the numerical value. This happens because the NE for the other environments is $\pm\gamma$ depending on if their optimal least squares attributions are above/below those of the middle two environments. This again is a stable and likely to be faithful choice, where also unidirectionality is preferred.

D Experimental Details

D.1 Dataset Details and Hyperparameter Specifications

We describe the datasets and the hyperparameters used for each. We set perturbation neighborhood sizes 10 (IRIS), 500 (MEPS), 100 (FMNIST-random), 500 (FMNIST-realistic), 100 (CIFAR10-random), 500 (CIFAR10-realistic), 100 (Rotten tomatoes) for generating local explanations. We also use 3, 10, 10, 10, 5 as exemplar neighborhood sizes to compute GI, CI and Υ metrics for the five datasets respectively. We also use $5-$sparse explanations for all cases except FMNIST and CIFAR10 with realistic perturbations where we follow MeLIME and generate a dense explanation using ridge penalty with penalty multiplier value of 0.001. The ℓ_∞ bound γ in Algorithm 1 is set as the maximum absolute value of linear coefficient computed by running LIME/MeLIME in the two individual environments. Please look at IRIS dataset first since it contains some of the common details used across others.

IRIS (Tabular): This dataset has 150 instances with four numerical features representing the sepal and petal width and length in centimeters. The task is to classify instances of Iris flowers into three species: *setosa*, *versicolor*, and *virginica*. A random forest classifier was trained with a train/test split of 0.8/0.2 and yielded a test accuracy of 93%. We provide local explanations for the prediction probabilities for class *setosa*. For both random and realistic perturbations, we use a perturbation neighborhood size of n. For random perturbations, we used the same approach followed by LIME and sample from a Gaussian around each data point. Realistic perturbations (with the same number n) were generated using KDEGen [8], a kernel density estimator (KDE) with the Gaussian kernel fitted on the training dataset to sample data around a sample point. For both random and realistic perturbations, we weight the neighborhood using a Gaussian kernel of width $\tau\sqrt{d}$, where d is the dimension of the feature vector and $\tau = \{0.05, 0.1, 0.25, 0.5, 0.75\}$, and this corresponded to kernel widths $\{0.1, 0.2, 0.5, 1.0, 1.5\}$. We also perform a weighted version of realistic selection where we use MAPLE [36] to assign weights to all the test examples and pick the top n weighted examples to use as the perturbation neighborhood. For random/realistic perturbations and realistic selection, the corresponding environments (of size n each) for LINEX are created by drawing k bootstrap samples where $k = \{2, 3, 4, 5\}$ in our experiments. We test for $n = \{10, 20, 30, 40, 50\}$ with this dataset.

Medical Expenditure Panel Survey (Tabular): The Medical Expenditure Panel Survey (MEPS) dataset is produced by the US Department of Health and Human Services. It is a collection of surveys of families of individuals, medical providers, and employers across the country. We choose *Panel 19* of the survey which consists of a cohort that started in 2014 and consisted of data collected over 5

rounds of interviews over $2014-2015$. The outcome variable was a composite utilization feature that quantified the total number of healthcare visits of a patient. The features used included demographic features, perceived health status, various diagnosis, limitations, and socioeconomic factors. We filter out records that had a utilization (outcome) of 0, and log-transformed the outcome for modeling. These pre-processing steps resulted in a dataset with 11136 examples and 32 categorical features. We train a random forest regressor that has a test R^2 of 0.325 in this dataset. We provide local explanations of the predictions. With MEPS, we do not use realistic perturbations since KDE and VAE generators do not work well with categorical data. Otherwise the setting is similar as IRIS data, except that we use $n = \{50, 100, 200, 300, 400, 500\}$. The kernel widths in this case were $\{0.28, 0.57, 1.41, 2.83, 4.24\}$. We use $k = \{2, 3, 4, 5\}$ for this dataset.

Fashion MNIST (Images): This dataset has 28×28 grayscale images of fashion articles with 60,000 train and 10,000 test samples. The task is to classify these into 10 classes corresponding to coat, shoe, and so on. A neural network trained with test accuracy of 87%. Explanations are generated for the prediction probabilities corresponding to the predicted class for each example. We choose 1000 test examples to generate explanations. Realistic perturbations were generated using VAEGen [8], a Variational Auto Encoder (VAE) fitted on the training dataset. For random perturbations, we chose n from $\{50, 100, 200, 300, 400, 500\}$ and kernel sizes were $\{0.43, 0.85, 2.14, 4.27, 6.41\}$. For realistic perturbations we chose n from $\{250, 500, 750, 1000\}$ and the kernel widths were $\{1.4, 2.8, 7.0, 14.0, 21.0\}$. We use $k = \{2, 3, 4, 5\}$ for this dataset.

CIFAR10 (Images): This dataset has 32 × 32 colored images belonging to 10 different classes. The dataset has 50,000 train and 10,000 test samples. The task is to classify these into 10 classes corresponding to dog, bird, and so on. A residual network with 18 units (ResNet18) was trained with test accuracy of $\sim 95\%$. Explanations are generated for the prediction probabilities corresponding to the predicted class for each example. We choose 1000 test examples to generate explanations. Realistic perturbations were generated using VAEGen [8], a Variational Auto Encoder (VAE) fitted on the training dataset. For random perturbations, we chose n from $\{50, 100, 200, 300, 400, 500\}$ and kernel sizes were $\{0.43, 0.85, 2.14, 4.27, 6.41\}$. For realistic perturbations we chose n from $\{250, 500, 750, 1000\}$ and the kernel widths were $\{1.4, 2.8, 7.0, 14.0, 21.0\}$. We use $k = \{2, 3, 4, 5\}$ for this dataset.

Rotten Tomatoes (Text): This dataset contains 10662 movie reviews from rotten tomatoes website along with their sentiment polarity, i.e., positive or negative reviews and the task is to classify the sentiment of the reviews into positive or negative. The review sentences were vectorized using CountVectorizer and TfidfTransformer and a sklearn Naive Bayes classifier was fitted on training dataset which yielded a test accuracy of 75%. Explanations are generated for the prediction probabilities corresponding to the predicted class for each example. Realistic perturbations were generated using Word2VecGen [8], wherein word2vec embeddings are first trained using the training corpus and new sentences are generated by randomly replacing a sentence word whose distance in the embedding space lies within the radius of the neighbourhood. For both random and realistic perturbations, n was chosen from $\{25, 50, 75, 100\}$. The kernel sizes were $\{0.42, 1.06, 2.12, 3.18\}$ for random perturbations (kernel size 0.21 resulted in numerical issues), and $\{0.21, 0.42, 1.06, 2.12, 3.18\}$ for realistic perturbations. We use $k = \{2, 3, 4, 5\}$ for this dataset.

E Results with All Datasets and Hyperparameter Combinations for Random and Realistic Perturbations

We present results with all hyperparameter combinations for random and realistic perturbations. Results for LIME with random perturbations (LIME), smoothed LIME (S-LIME), LINEX with random perturbations (LINEX/rand), MeLIME (MeLIME), LINEX with MeLIME-like realistic neighborhoods (LINEX/real), MAPLE (MAPLE), LINEX with MAPLE-like realistic neighborhoods (LINEX/mpl) are presented in figures 5-19. The legend for these figures are given in Figure 4.

For the five datasets, we perform ablations by varying one of perturbation neighborhood size (Figures 5-9), number of environments (Figures 10-14), and kernel width (Figures 15-19). Each point in these figures are averaged over all possible values for the two parameters that are not ablated. For example, each point in Figure 5 is averaged over all possible values for kernel widths and number of

environments for a given perturbation neighborhood size. Standard errors of the mean are also plotted in the same color with lesser opacity. Lower values of Infidelity (INFD), Generalized Infidelity (GI), Coefficient Inconsistency (CI) are better whereas for Unidirectionality (Υ) and Class Attribution Consistency (CAC) higher values are better.

Figures 5-9 show ablations with respect to perturbation neighborhood sizes. Considering all datasets, the stability/recourse metrics (CI, Υ, CAC) are clearly better for LINEX compared to its counterparts. For LINEX methods (LINEX/rand, LINEX/real, LINEX/mpl), the metrics get better or stays approximately the same generally as perturbation neighborhood size increases keeping with the intuition that larger perturbation neighborhood sizes should produce explanations that are more stable in the exemplar neighborhood. Υ for FMNIST and CIFAR10 are already good for small perturbation neighborhood sizes possibly because of the quality of MeLIME perturbations.

Turning to the fidelity metrics (INFD and GI) in tabular datasets, we see that the results still favor LINEX, but less heavily compared to the stability/recourse metrics. This is in line with what we observe in Table 2. In IRIS and MEPS, LINEX is close to or outperforms the corresponding baselines in the GI measure (except for LINEX/mpl with MEPS). This gap closes a bit with INFD, but we note that GI is a better measure since it estimates how faithful explanations are in a exemplar neighborhood. With the text dataset, LINEX variants are slightly more favored, whereas with the image dataset, the baselines have an edge.

Considering Figures 10-14, we see that variations are less stark with respect to number of environments overall for LINEX variants. Note that except for S-LIME, other baselines do not use multiple environments, and hence stay constant. The slight variations in MAPLE are due to the effect of random seeds. In the stability/recourse metrics, again LINEX variants emerge as the clear winner across datssets. With the faithfulness metrics (GI and INFD), in the text dataset, LINEX variants generally perform better, whereas the baselines have a better performance in the image dataset.

Finally, we study the variation of the performance measures with respect to kernel width in Figures 15-19. We see that the stability/recourse metrics flatten out in all cases with large kernel widths. This behaviour holds true for faithfulness metrics (GI and INFD) as well except in some cases. GI and INFD measures also increase before they flatten out since the fit becomes poorer at larger kernel widths. The stability/recourse metrics become better or remain approximately the same since explanations generally improve or preserve their stability properties as kernel widths increase. Note that very small kernel widths can lead to unexpected behavior that does not fit the trend as seen with the tabular datasets since explanations can become hyper-local. MAPLE and LINEX/mpl stay the same at different kernel widths since they use a different weighting scheme. As with other ablations, we see that LINEX variants are similar or better in stability/recourse metrics overall, while with the faithfulness metrics the results are more mixed.

Note that we do not compute MeLIME perturbations with MEPS since KDE and VAE generators do not work well with categorical data, and do not use compute CAC since the task is regression. Further, the features used in explanations for different test examples are not comparable for random perturbations with FMNIST, CIFAR10 and Rotten Tomatoes, hence we cannot compute CAC for those cases as well. This explains the missing curves/plots.

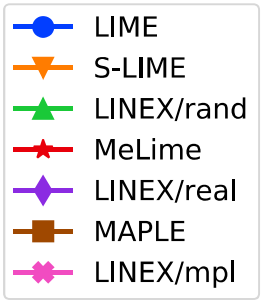
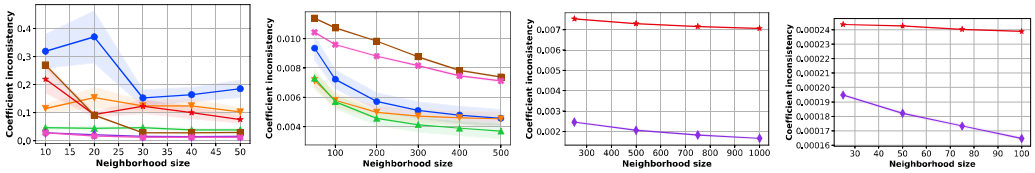
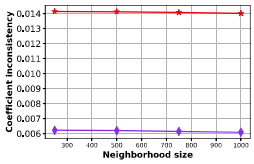


Figure 4: Legend for figures 5-19

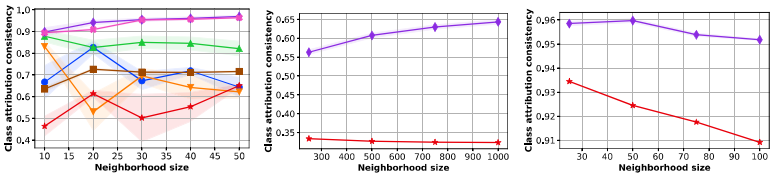


(a) Iris (b) MEPS (c) FMNIST (d) Rotten Tomatoes



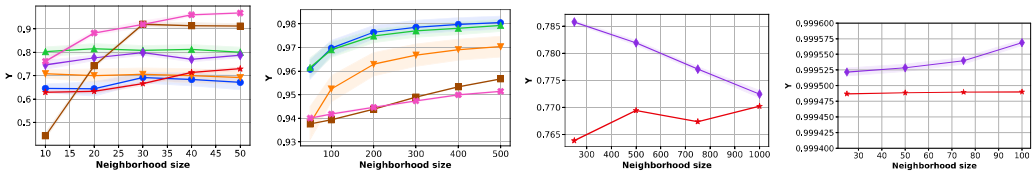
(e) CIFAR10

Figure 5: Coefficient inconsistency (CI) vs. Perturbation neighborhood size.

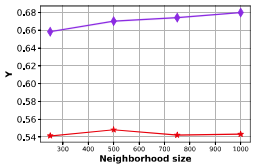


(a) Iris (b) FMNIST (c) Rotten Tomatoes

Figure 6: Class attribution consistency (CAC) vs. Perturbation neighborhood size.

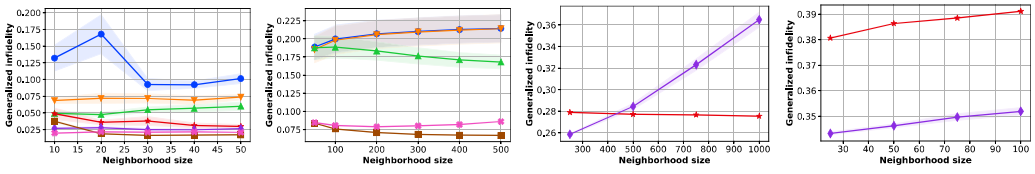


(a) Iris (b) MEPS (c) FMNIST (d) Rotten Tomatoes

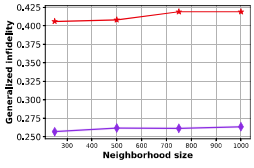


(e) CIFAR10

Figure 7: Unidirectionality (Υ) vs. Perturbation neighborhood size.

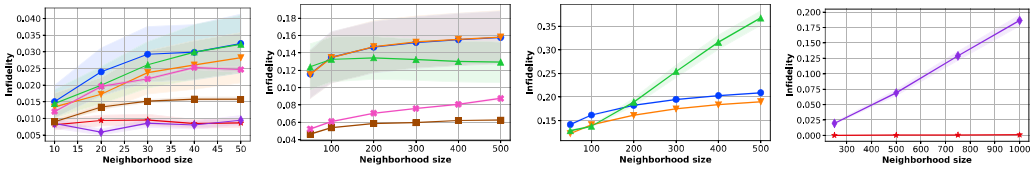


(a) Iris (b) MEPS (c) FMNIST (d) Rotten Tomatoes

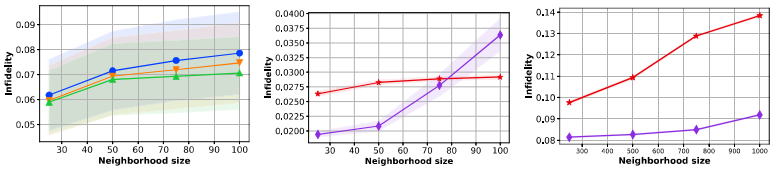


(e) CIFAR10

Figure 8: Generalized infidelity (GI) vs. Perturbation neighborhood size.

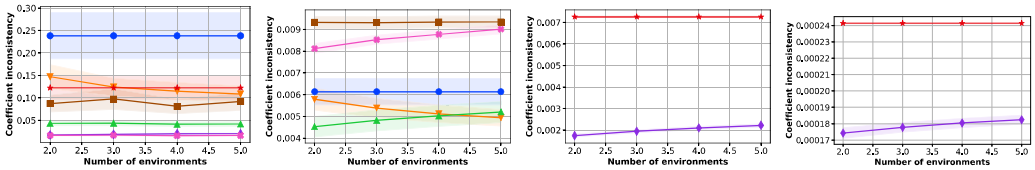


(a) Iris (b) MEPS (c) FMNIST (random) (d) FMNIST (realistic)

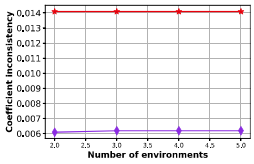


(e) Rotten Tomatoes (random) (f) Rotten Tomatoes (realistic) (g) CIFAR10

Figure 9: Infidelity (INFD) vs. Perturbation neighborhood size.

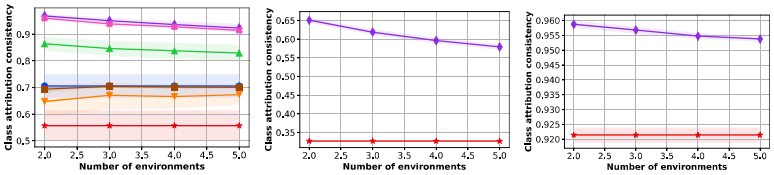


(a) Iris (b) MEPS (c) FMNIST (d) Rotten Tomatoes



(e) CIFAR10

Figure 10: Coefficient inconsistency (CI) vs. Number of environments.



(a) Iris (b) FMNIST (c) Rotten Tomatoes

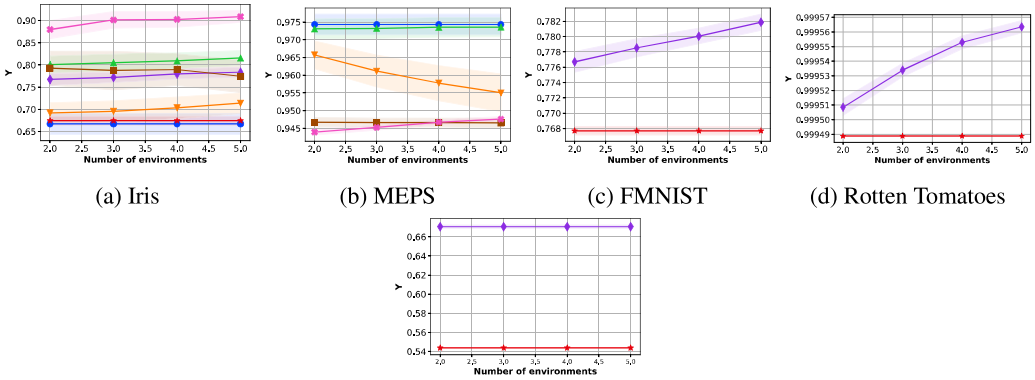
Figure 11: Class attribution consistency (CAC) vs. Number of environments.

F Example Feature Attributions in Text Data: MeLIME vs LINEX

Below we see sample attributions by the two methods along with the magnitude of the attributions. Attribution magnitudes are printed with a precision of 10^{-3} and shown along with the corresponding words in descending order.

F.1 Positive Sentiment

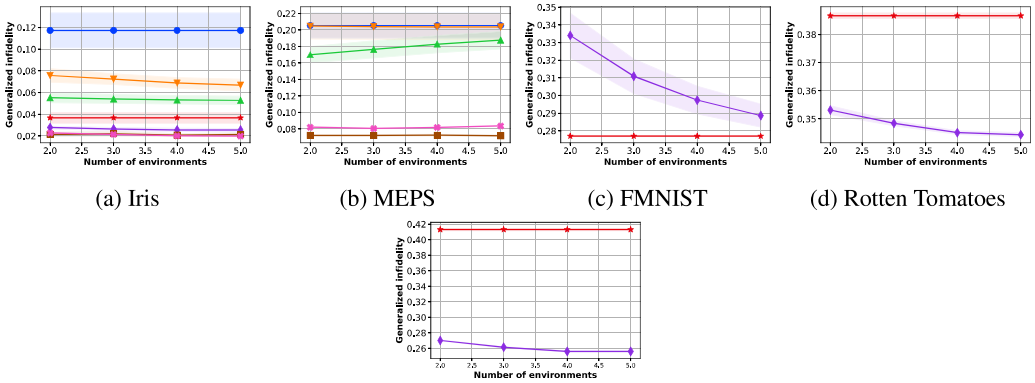
```
enticing and often funny documentary .
MeLIME: documentary funny and enticing often
LINEX : documentary funny often enticing and
MeLIME: 0.517 0.446 0.333 0.317 0.311
```



(a) Iris (b) MEPS (c) FMNIST (d) Rotten Tomatoes

(e) CIFAR10

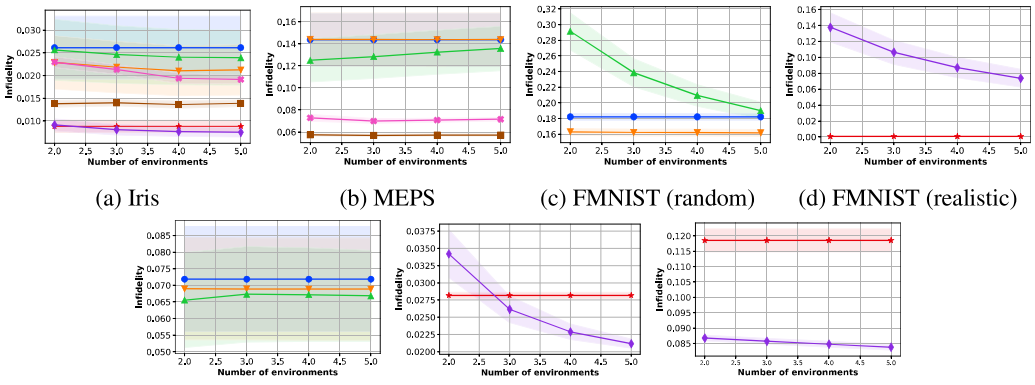
Figure 12: Unidirectionality (Υ) vs. Number of environments.



(a) Iris (b) MEPS (c) FMNIST (d) Rotten Tomatoes

(e) CIFAR10

Figure 13: Generalized infidelity (GI) vs. Number of environments.

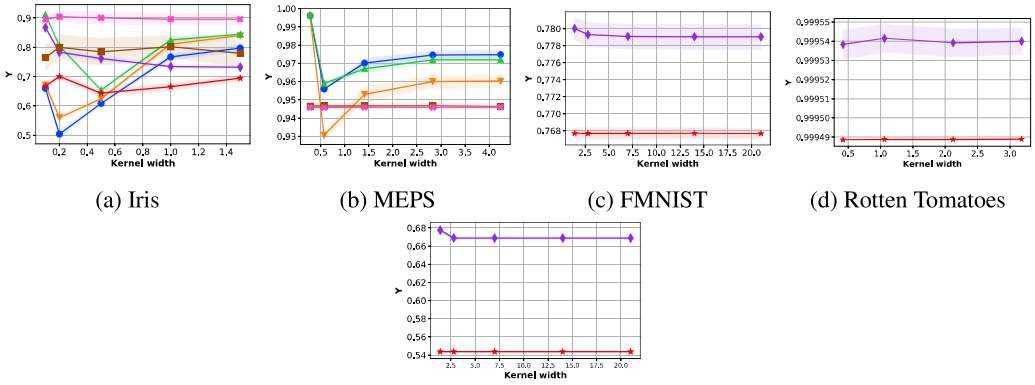


(a) Iris (b) MEPS (c) FMNIST (random) (d) FMNIST (realistic)

(e) Rotten Tomatoes (random) (f) Rotten Tomatoes (realistic) (g) CIFAR10

Figure 14: Infidelity (INFD) vs. Number of environments.

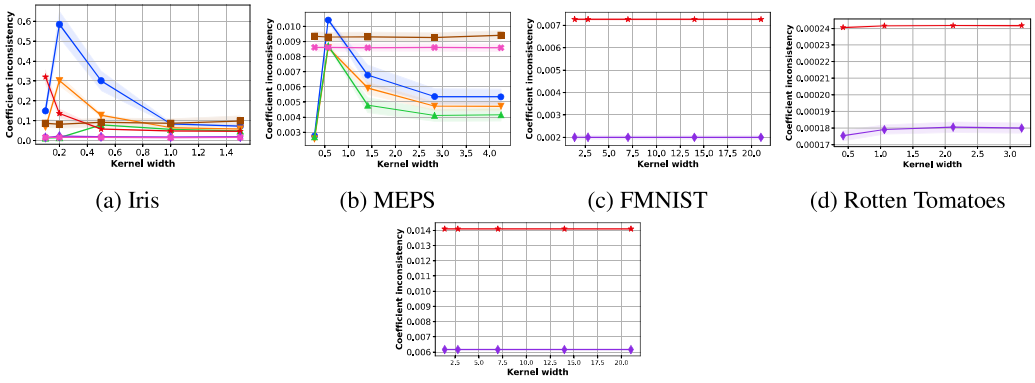
```
LINEX : 0.416 0.377 0.342 0.331 0.330

one-of-a-kind near-masterpiece .
MeLIME: kind near masterpiece
LINEX : masterpiece kind one
MeLIME: 0.832 0.695 0.182
LINEX : 0.712 0.384 0.381
```



(a) Iris (b) MEPS (c) FMNIST (d) Rotten Tomatoes

(e) CIFAR10

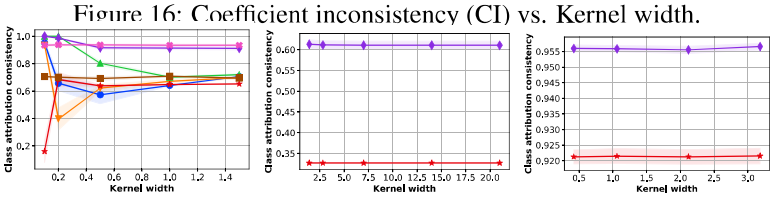
Figure 15: Unidirectionality (Υ) vs. Kernel width.



(a) Iris (b) MEPS (c) FMNIST (d) Rotten Tomatoes

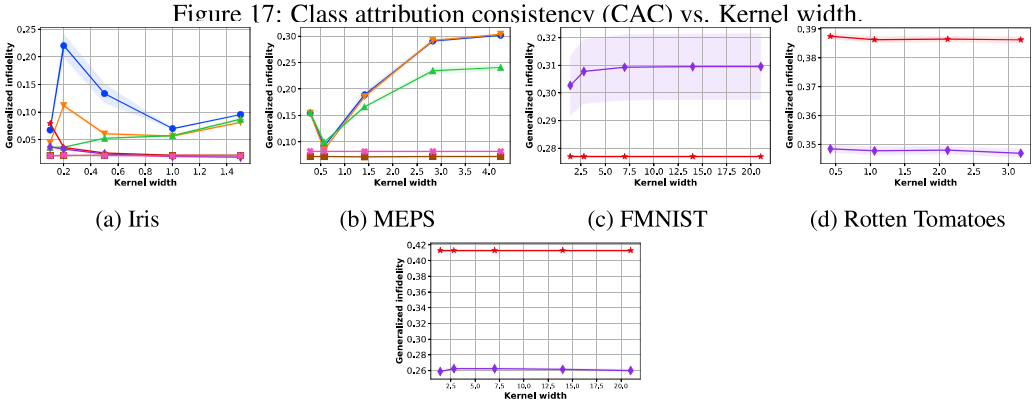
(e) CIFAR10

Figure 16: Coefficient inconsistency (CI) vs. Kernel width.



(a) Iris (b) FMNIST (c) Rotten Tomatoes

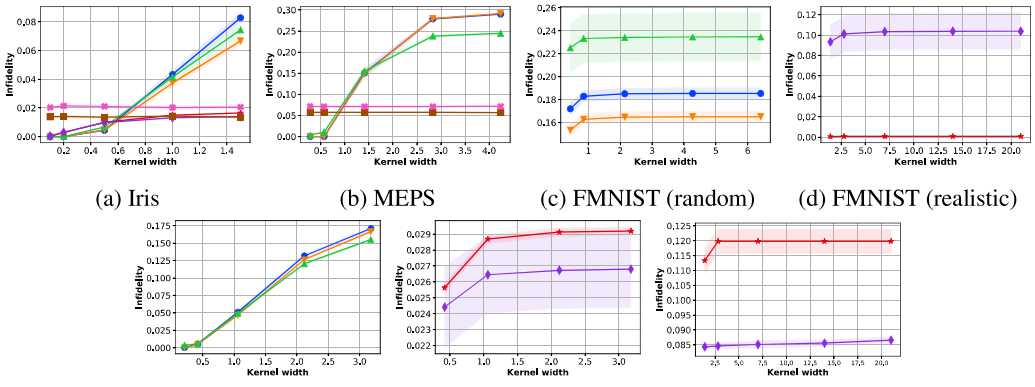
Figure 17: Class attribution consistency (CAC) vs. Kernel width.



(a) Iris (b) MEPS (c) FMNIST (d) Rotten Tomatoes

(e) CIFAR10

Figure 18: Generalized infidelity (GI) vs. Kernel width.



(a) Iris (b) MEPS (c) FMNIST (random) (d) FMNIST (realistic)

(e) Rotten Tomatoes (random) (f) Rotten Tomatoes (realistic) (g) CIFAR10

Figure 19: Infidelity (INFD) vs. Kernel width.

```
a fast , funny , highly enjoyable movie .
MeLIME: enjoyable highly funny fast movie
LINEX : enjoyable highly fast funny movie
MeLIME: 0.550 0.432 0.412 0.389 0.198
LINEX : 0.409 0.389 0.372 0.350 0.326

ferrara's strongest and most touching movie of recent years .
MeLIME: touching years most strongest and
LINEX : touching most recent strongest and
MeLIME: 0.735 0.490 0.450 0.443 0.427
LINEX : 0.490 0.488 0.450 0.444 0.407

saved from being merely way-cool by a basic , credible compassion .
MeLIME: cool basic credible merely from
LINEX: cool credible merely compassion from
MeLIME: 1.514 0.050 0.040 0.029 0.026
LINEX : 0.358 0.308 0.304 0.299 0.293

really quite funny .
MeLIME: funny quite really
LINEX : funny quite really
MeLIME: 0.559 0.417 0.233
LINEX : 0.462 0.368 0.275

spare yet audacious . . .
MeLIME: spare yet audacious
LINEX : audacious spare yet
MeLIME: 0.626 0.447 0.395
LINEX : 0.501 0.431 0.422

an engrossing and infectiously enthusiastic documentary .
MeLIME: engrossing documentary and enthusiastic an
LINEX : engrossing documentary an enthusiastic and
MeLIME: 0.593 0.455 0.358 0.354 0.333
LINEX : 0.461 0.407 0.374 0.357 0.350

a wildly funny prison caper .
MeLIME: funny caper wildly prison
LINEX : funny caper prison wildly
MeLIME: 0.541 0.364 0.214 0.193
```

```
LINEX : 0.403 0.335 0.245 0.239

this charming but slight tale has warmth , wit
and interesting characters compassionately portrayed .
MeLIME: charming compassionately and interesting portrayed
LINEX : charming compassionately has tale portrayed
MeLIME: 0.690 0.507 0.456 0.444 0.424
LINEX : 0.464 0.435 0.431 0.430 0.429

thoughtful , provocative and entertaining .
MeLIME: thoughtful entertaining and provocative
LINEX : thoughtful entertaining and provocative
MeLIME: 0.612 0.517 0.402 0.395
LINEX : 0.505 0.461 0.415 0.404

the film is quiet , threatening and unforgettable .
MeLIME: quiet unforgettable and film the
LINEX : unforgettable quiet film and is
MeLIME: 0.597 0.483 0.412 0.325 0.303
LINEX : 0.421 0.416 0.388 0.378 0.338

a moving tale of love and destruction in unexpected places , unexamined lives .
MeLIME: unexpected moving love tale lives
LINEX : moving unexpected places lives in
MeLIME: 0.692 0.662 0.577 0.538 0.499
LINEX : 0.538 0.530 0.521 0.513 0.501

though frodo's quest remains unfulfilled , a hardy group of
determined new zealanders has proved its creative mettle .
MeLIME: creative group proved has new
LINEX : creative quest its proved determined
MeLIME: 0.602 0.441 0.424 0.402 0.393
LINEX : 0.410 0.392 0.390 0.385 0.381
```

F.2 Negative Sentiment

```
originality is sorely lacking .
MeLIME: lacking sorely is originality
LINEX : lacking sorely originality is
MeLIME: 0.543 0.381 0.296 0.278
LINEX : 0.430 0.356 0.314 0.271

an ugly , pointless , stupid movie .
MeLIME: stupid pointless ugly movie an
LINEX : stupid pointless ugly movie an
MeLIME: 0.543 0.499 0.385 0.365 0.276
LINEX : 0.446 0.411 0.373 0.360 0.350

so devoid of pleasure or sensuality that it cannot even be dubbed hedonistic .
MeLIME: devoid even be dubbed of
LINEX : devoid so dubbed be cannot
MeLIME: 0.666 0.416 0.413 0.372 0.344
LINEX : 0.400 0.392 0.387 0.380 0.368

neither revelatory nor truly edgy--merely crassly flamboyant
and comedically labored .
MeLIME: edgy neither nor labored revelatory
LINEX : edgy neither nor labored truly
MeLIME: 1.256 0.338 0.277 0.204 0.021
```

```
LINEX : 0.439 0.398 0.398 0.369 0.349

occasionally funny , sometimes inspiring , often boring .
MeLIME: boring occasionally inspiring sometimes often
LINEX : boring occasionally sometimes often inspiring
MeLIME: 0.669 0.242 0.218 0.210 0.182
LINEX : 0.377 0.266 0.266 0.250 0.236

a cumbersome and cliche-ridden movie greased
with every emotional device known to man .
MeLIME: cliche every device movie with
LINEX : cliche every man cumbersome emotional
MeLIME: 0.695 0.449 0.327 0.280 0.268
LINEX : 0.385 0.361 0.354 0.349 0.309

ponderous , plodding soap opera disguised as a feature film .
MeLIME: plodding soap ponderous opera disguised
LINEX : plodding soap film ponderous feature
MeLIME: 0.579 0.522 0.421 0.408 0.382
LINEX : 0.442 0.440 0.418 0.406 0.377

kitschy , flashy , overlong soap opera .
MeLIME: soap flashy opera overlong kitschy
LINEX : soap flashy opera overlong kitschy
MeLIME: 0.499 0.397 0.391 0.358 0.230
LINEX : 0.389 0.362 0.360 0.346 0.300

[a] poorly executed comedy .
MeLIME: poorly comedy executed
LINEX : poorly comedy executed
MeLIME: 0.653 0.348 0.257
LINEX : 0.502 0.335 0.309

a bad movie that happened to good actors .
MeLIME: bad happened movie to that
LINEX : bad happened to movie actors
MeLIME: 0.692 0.396 0.371 0.367 0.242
LINEX : 0.442 0.384 0.367 0.361 0.344

a complete waste of time .
MeLIME: waste complete time of
LINEX : waste complete time of
MeLIME: 0.614 0.425 0.313 0.247
LINEX : 0.480 0.381 0.348 0.278

don't waste your money .
MeLIME: waste money don your
LINEX : waste money don your
MeLIME: 0.592 0.497 0.408 0.309
LINEX : 0.483 0.450 0.411 0.337

witless and utterly pointless .
MeLIME: pointless witless and utterly
LINEX : pointless witless utterly and
MeLIME: 0.652 0.491 0.263 0.245
LINEX : 0.506 0.444 0.311 0.269
```

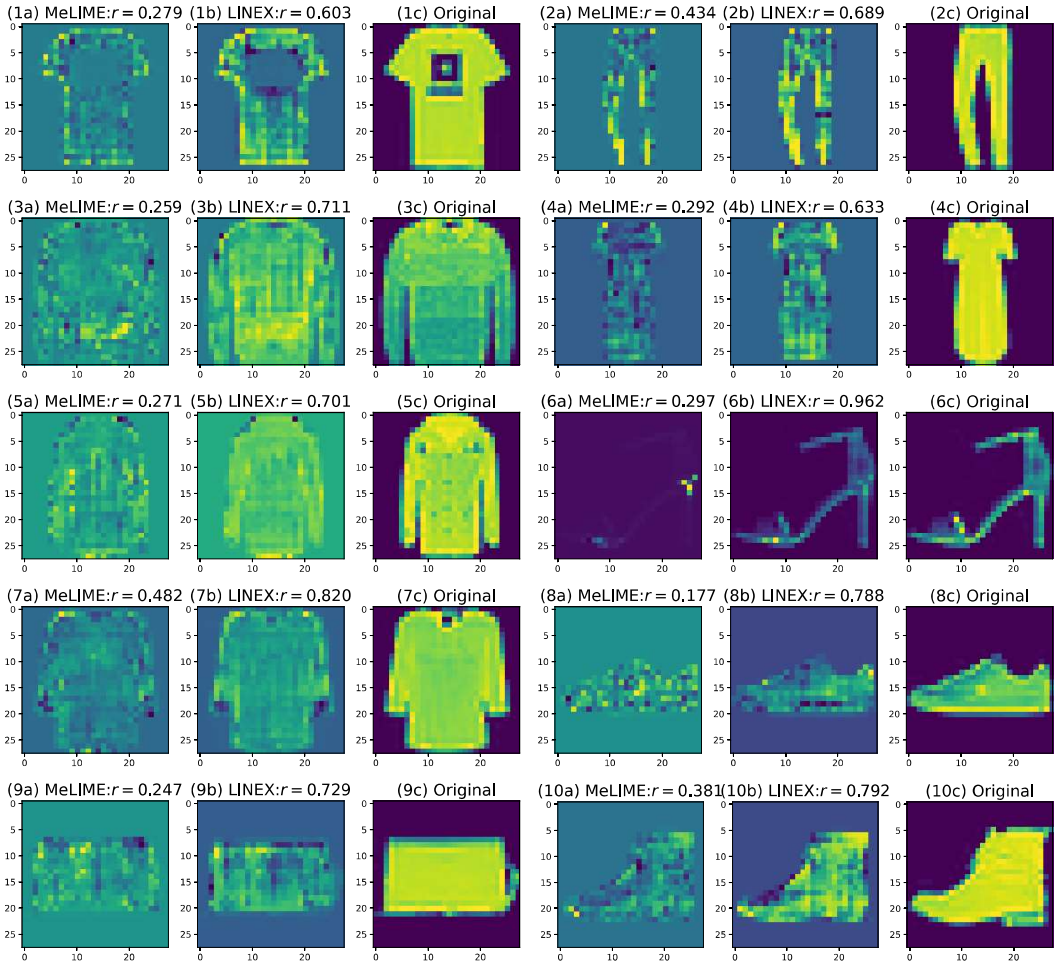


Figure 20: Results using individual samples for realistic perturbations for FMNIST dataset for all classes:1-10 (*T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag* and *Ankle boot*). (a) MeLIME feature attributions for an image. (b) LINEX feature attributions for an image. (c) Original image in the class. The r values show Pearson's correlation between feature attributions and the original image from the respective class. We observe that LINEX attributions/explanations exhibit significantly higher correlation with the original image belonging to a particular class (i.e. high CAC).

G Example Feature Attributions in Image Data: MeLIME vs LINEX

We show feature attributions for individual example images with MeLIME and LINEX with MeLIME perturbations in Figure 20. In Figure 21 we show class-wise mean feature attributions along with mean images. In Figure 22, we see examples from CIFAR10. LINEX explanations seem to provide more meaningful feature attributions.

H Results for All Methods Including SHAP

In Table 4, we provide the results for SHAP along with all methods for easy comparison. Note that SHAP does not have standard errors since it is computed only once per test point. The INFD values for SHAP are miniscule since SHAP values add up to the predictions by definition. In order to compute GI, CI, Υ, CAC, we convert the SHAP values to SHAP attributions [5] first and follow the same approach used by other explanation methods.

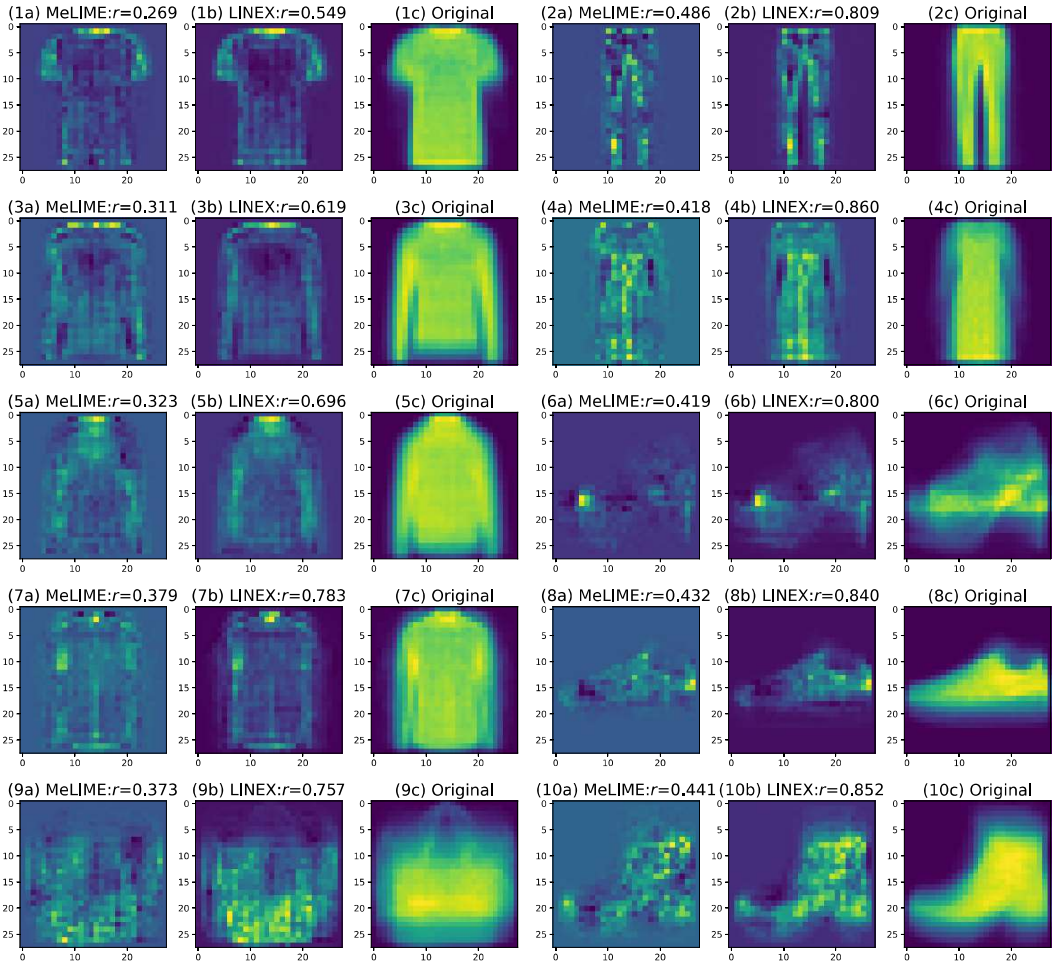


Figure 21: Results using realistic perturbations for FMNIST dataset with mean feature importances for all classes:1-10 (*T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag* and *Ankle boot*). (a) Mean feature attributions of all images in the class using MeLIME. (b) Mean feature attributions of all images in the class using LINEX. (c) Mean of all images in the class. The r values show Pearson's correlation between average feature attributions and mean of the original images from the respective classes. We observe that LINEX explanations/attributions exhibit significantly higher correlation with the original images belonging to a particular class (i.e. high CAC).

I Error Analysis of LINEX

We perform error analysis for LINEX to gain better understanding about the method. We choose FMNIST dataset for doing this since, LINEX/real under performs MeLIME in terms of the INFD measure here (see Table 2) more heavily compared to other datasets and so we wanted to investigate the reasons for this. This also happens to be one of the higher dimensional datasets that is intuitive to visualize and understand.

We start by observing that even though LINEX/real underperforms in the INFD metric, the gap is not so great in the GI metric, which suggests that MeLIME may be overfitting explanations here. We also note that in terms of CI, Υ, and CAC metrics, LINEX/real clearly outperforms MeLIME.

We now choose a sample of images from the dataset where LINEX/real has highest instance-level infidelity numbers and display them in Figure 23. Just looking at the explanations and the corresponding original images visually, it is evident that LINEX/real highlights the prominent features like sleeves and collar in a shirt, handles of the bags, outlines of the boots/shoes, even though the infidelity values are high. However, MeLIME misses out on some of these prominent features and

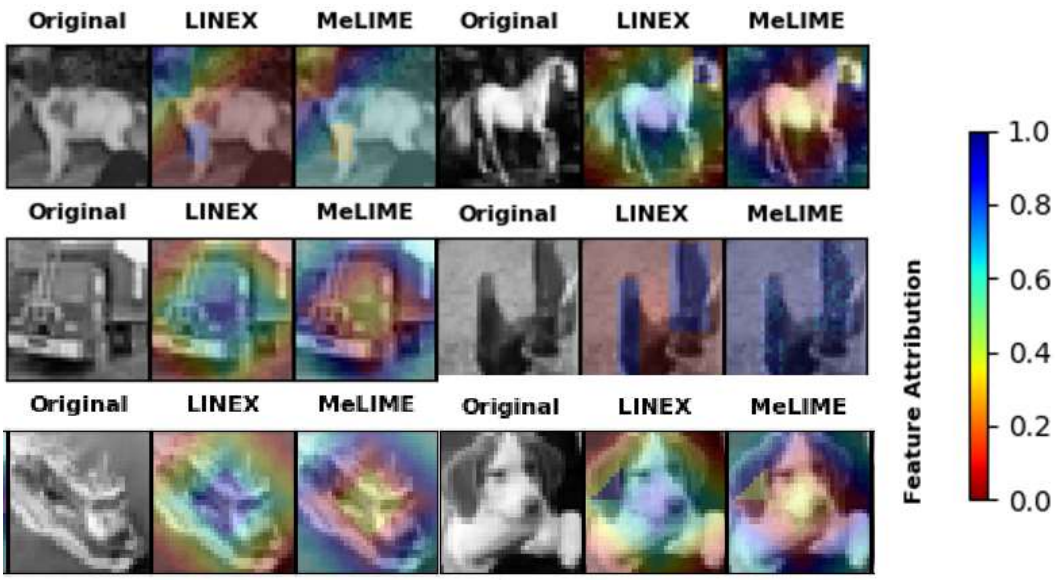


Figure 22: Results using realistic perturbations for CIFAR10 dataset. We see above images of a dog, a horse, a truck, a bird, a boat and a dog again randomly selected from CIFAR10. The original images are greyed out here so that the (normalized) attributions are clearly visible. As can be seen LINEX attributions seem to consistently focus on salient features as compared to MeLIME. For example for the first dog image we highlight the head, ears and leg, while MeLIME focuses more on the neck and some of the background. For horse too LINEX focuses on head and body, while MeLIME focuses on the legs and neck. For truck both seem to focus on important features. For bird LINEX hones in on the wings, while MeLIME although giving importance to wings also attributes some of the background. The boat image LINEX focuses on the center of the boat, while Melime on the edges and some of the water around the boat. For the dog face image LINEX focuses on the nose, eyes and ears, while Melime focuses on the ears and neck.

focuses only on optimizing the local fit. The fact that LINEX zeroes in on important features also provides additional evidence for the closeness of GI metrics between the two methods, and the better performance of LINEX/real with CI, Υ, and CAC metrics.

This conclusion is also verified when we look at the performance of LINEX at a class level. In Figure 24, we see two classes one where the infidelity of LINEX is low (i.e. Trousers class) and the other where its infidelity is high (i.e Shirt class). As can be seen since the Trousers class has examples with less superfluous features (viz. varied designs) focusing on which might reduce infidelity but are not critical for determination of the class, LINEX does better in terms of infidelity on the prior. However, although infidelity is higher for the latter Shirt class it does much better on other metrics such as GI, CAC, CI and Υ indicating that LINEX truly focuses on robust features.

J Ablation Analysis of Important Features for Various Explanation Methods

We wanted to analyze the most challenging case for us in the reported experiments which is on the FMNIST dataset where we are more worse than MeLIME in terms of INFD than any of the other setups. We thus assess if the features deemed important - those with the largest coefficients - by the explanation methods are indeed important for the black box model to make their predictions. To assess this, we set the we set a fraction of features (pixel values) corresponding to the top coefficients of MeLIME and LINEX/realistic to a baseline value and run the modified images again through the black box model - this is what we mean by ablation here. The baseline value here was chosen to be -1 since that is the value of the background pixels. We then used two measures to assess the quality of explanations - higher values being better for both. The first measure is mean absolute error between the predicted scores before and after ablation, corresponding to the original predicted class. The second measure is the fraction of images that changed their predicted class after ablation. We

Table 4: Comparing the different methods (including SHAP) using metrics infidelity (INFD), generalized infidelity (GI), coefficient inconsistency (CI), class attribution consistency (CAC) and unidirectionality (Υ).

Dataset	*Method*	INFD ↓	GI ↓	CI ↓	Υ ↑	CAC ↑
IRIS	LIME	0.015 ± 0.011	0.132 ± 0.042	0.319 ± 0.132	0.646 ± 0.040	0.667 ± 0.167
	S-LIME	0.015 ± 0.010	0.077 ± 0.011	0.143 ± 0.045	0.704 ± 0.037	0.878 ± 0.034
	LINEX/rand	0.013 ± 0.009	$\mathbf{0.052 \pm 0.008}$	$\mathbf{0.044 \pm 0.013}$	$\mathbf{0.802 \pm 0.043}$	$\mathbf{0.921 \pm 0.042}$
	NB/rand	0.040 ± 0.010	0.067 ± 0.003	0.319 ± 0.132	0.646 ± 0.040	0.667 ± 0.167
	MeLIME	0.008 ± 0.003	0.049 ± 0.018	0.219 ± 0.108	0.629 ± 0.013	0.464 ± 0.100
	LINEX/real	0.009 ± 0.003	$\mathbf{0.029 \pm 0.003}$	$\mathbf{0.024 \pm 0.002}$	$\mathbf{0.744 \pm 0.044}$	$\mathbf{0.942 \pm 0.023}$
	NB/real	0.058 ± 0.022	0.034 ± 0.000	0.219 ± 0.108	0.629 ± 0.013	0.464 ± 0.100
	MAPLE	0.009 ± 0.001	0.038 ± 0.004	0.261 ± 0.033	0.458 ± 0.032	0.586 ± 0.035
	LINEX/mpl	0.013 ± 0.000	$\mathbf{0.020 \pm 0.000}$	$\mathbf{0.026 \pm 0.002}$	$\mathbf{0.694 \pm 0.008}$	$\mathbf{0.929 \pm 0.004}$
	SHAP	0.007	0.197	0.248	0.664	0.524
MEPS	LIME	0.158 ± 0.066	0.214 ± 0.041	0.005 ± 0.001	0.981 ± 0.006	NA
	S-LIME	0.158 ± 0.066	0.214 ± 0.042	0.005 ± 0.001	0.974 ± 0.008	
	LINEX/rand	$\mathbf{0.130 \pm 0.052}$	$\mathbf{0.164 \pm 0.021}$	0.003 ± 0.001	0.979 ± 0.006	
	NB/rand	0.275 ± 0.062	0.311 ± 0.079	0.005 ± 0.001	0.981 ± 0.006	
	MAPLE	$\mathbf{0.063 \pm 0.000}$	$\mathbf{0.067 \pm 0.000}$	0.007 ± 0.000	0.957 ± 0.000	NA
	LINEX/mpl	0.098 ± 0.001	0.094 ± 0.001	0.007 ± 0.000	0.950 ± 0.000	
	SHAP	0.000	0.091	0.009	0.940	NA
FMNIST	LIME	0.162 ± 0.003	NA	NA	NA	NA
	S-LIME	0.142 ± 0.003				
	LINEX/rand	0.149 ± 0.002				
	NB/rand	0.207 ± 0.000				
	MeLIME	$\mathbf{0.001 \pm 0.000}$	$\mathbf{0.277 \pm 0.000}$	0.007 ± 0.000	0.769 ± 0.000	0.327 ± 0.000
	LINEX/real	0.100 ± 0.002	0.304 ± 0.001	0.002 ± 0.000	$\mathbf{0.780 \pm 0.000}$	$\mathbf{0.649 \pm 0.001}$
	NB/real	0.017 ± 0.000	0.446 ± 0.000	0.007 ± 0.000	0.769 ± 0.000	0.327 ± 0.000
	SHAP	0.000	1.962	0.589	0.551	0.038
CIFAR10	LIME	0.191 ± 0.005	NA	NA	NA	NA
	S-LIME	0.185 ± 0.002				
	LINEX/rand	0.186 ± 0.002				
	NB/rand	0.208 ± 0.001				
	MeLIME	0.100 ± 0.003	0.412 ± 0.007	0.014 ± 0.000	0.546 ± 0.003	NA
	LINEX/real	0.090 ± 0.005	$\mathbf{0.279 \pm 0.001}$	$\mathbf{0.006 \pm 0.000}$	$\mathbf{0.679 \pm 0.004}$	
	NB/real	0.103 ± 0.002	0.398 ± 0.004	0.014 ± 0.000	0.546 ± 0.003	
	SHAP	0.003	1.376	0.398	0.512	NA
Rotten Tomatoes	LIME	0.079 ± 0.036	NA	NA	NA	NA
	S-LIME	0.075 ± 0.035				
	LINEX/rand	0.069 ± 0.032				
	NB/rand	0.241 ± 0.007				
	MeLIME	$\mathbf{0.029 \pm 0.001}$	0.391 ± 0.000	0.000 ± 0.000	0.999 ± 0.000	0.909 ± 0.000
	LINEX/real	0.053 ± 0.000	$\mathbf{0.361 \pm 0.000}$	0.000 ± 0.000	1.000 ± 0.000	$\mathbf{0.953 \pm 0.001}$
	NB/real	0.035 ± 0.000	0.535 ± 0.000	0.000 ± 0.000	0.999 ± 0.000	0.909 ± 0.000
	SHAP	0.000	0.384	0.008	0.999	0.015

see from Figure 25 that LINEX/realistic substantially outperforms MeLIME in both these measures, clearly demonstrating the relevance of features chosen by our method to the black box.

K Error Analysis of LINEX based on Ablation

Highlighting stable features for examples near non-linearities is a key strength of LINEX. However, in some cases for examples near class boundaries it may ignore sensitive features as we show in this demonstration.

In Figure 26, we show 6 examples that are appear to be close to class boundaries. We ablate pixels corresponding to top 15% of important features chosen by MeLIME and LINEX/realistic using the approach discussed in Section J. Ablation based on MeLIME importances meaningfully changes classes, whereas ablation by LINEX importances does not. The changes in prediction for MeLIME ablation for the six images are respectively from *Dress* to *Trouser*, *Sneaker* to *Sandal*, *Pullover* to *Dress*, *Sneaker* to *Sandal*, *Bag* to *Pullover*, and *Sneaker* to *Sandal*. The new class assignment looks

reasonable looking at the ablated images. We also see that the changes in class probabilities for the original class (p) are much higher after MeLIME ablation compared to LINEX/realistic ablation.

MeLIME ablated images for the first example has structures that look like trouser legs, for the second, fourth and sixth examples the area around the heel is more open making the original sneaker look like a sandal, for the third example, there is a hole in the hooded part of the pullover making it resemble a dress. The fifth example is classified as a pullover possibly because of the elongated structures on the sides that look like hands.

Note that such cases of LINEX under performing are rare though as is confirmed by its superior performance in Figure 25.

L Understanding Behavior of LIME and LINEX with Synthetic Data

We consider explaining the behavior of a function of two variables x and y with Class 1 sandwiched between Class 0 (see Figure 27). The third (or vertical) axis denotes the probability of being in Class 1. Clearly, x is the only important feature here that determines the class label.

From Figure 27 (left), we see that the LIME (here MeLIME would be the same as LIME since the space is flat and all points are realistic) feature attributions at points a, b, and c will provide importance to x feature for small as well as large kernel width (1 and 2 respectively) neighborhoods. For point c, in the interior of the Class 0, the attributions are stable across kernel widths. However for points a and b close to the boundary of classes, the attributions for small kernel width and large kernel width neighborhoods differ significantly along the x direction. This shows the instability of LIME explanations near boundaries of classes for different kernel widths.

In contrast in Figure 27 (right), we see that the LINEX explanation constructed for the two kernel widths provides stable feature attributions for all points a, b, c. For a and b, LINEX will conservatively pick a smaller feature attribution along the x direction since the function changes rapidly in its neighborhood. As such though LINEX will still pick the feature in the x direction in this scenario.

M Variation of feature attributions with γ

Based on the proof of Theorem 1, if for a feature the optimal attributions have opposite sign for each of the two environments, then γ can be made arbitrarily small (except 0) or large and the output of Algorithm 1 should still be the same which is 0 as the Nash Equilibrium is $\pm\gamma$. If the optimal attributions are the same sign then we should still get the same output from Algorithm 1 as long as $\gamma \geq \min(|w_{1i}|, |w_{2i}|)$ since the attribution from our algorithm is the minimum of those values. When $\gamma < \min(|w_{1i}|, |w_{2i}|)$ then the feature attributions will smoothly reduce as γ reduces.

We demonstrate this behavior in Figure 28 using an example from the IRIS dataset with random perturbations using the same setting as in Section 5. In the experiments in Section 5, we set $\gamma = 0.329$ which is the maximum absolute value based on a linear fit to each environment. As γ increases beyond 0.329, the attributions are unchanged demonstrating robustness. Same holds true while reducing γ up to 0.165 beyond which we see smooth reduction in the attribution values. Qualitatively, similar behavior is seen for other examples too. Because we set γ pessimistically (ignoring constraints) to a high value, we can expect our reported performances in the paper to be robust across many values of γ.

N Convergence of LINEX procedure and comparisons

We demonstrate based on a synthetic example how Algorithm 1 and provides a unidirectional explanation. We generate synthetic data using a function in $\mathbb{R}^2$ (Figure 29(left)). The function gently rises with increasing y values, and along x it is flat first, then rises abruptly and then falls gradually. We want to obtain robust attributions of this function at the point $x = 1.0, y = 0.0$, which is close to the end of the rising edge along x direction.

As we can imagine, since the slope changes abruptly along x direction near the point, it should be ideally excluded from an explanation intended towards recourse based on a linear proxy. Otherwise, the explanation will not generalize in the neighborhood of this point. On the other hand, the y direction should be included since the function changes smoothly along y throughout.

To generate explanations We first create two environments centered at the example to explain with variances 0.5 and 2.0. Now independently fitting to these environments leads to feature attributions that are $\{-0.033, 0.098\}$ and $\{0.084, 0.102\}$. Appending the two environments the attributions are $\{0.029, 0.095\}$, whereas with LINEX, the attributions would be $\{0.0, 0.093\}$. Thus, LINEX effectively eliminates the feature with high variability or abrupt changes. The behavior of the coefficients for each environment as LINEX converges is shown in Figure 29(right). As such, one can also see the convergence is fast.

O Limitations

Like any other posthoc explainable AI method there is no way to surely say that LINEX exactly reflects the true reasoning behind a black box classifier in arbitrary applications. It also is somewhat slower than LIME as shown in section A given the game theoretic nature of the algorithm, where its stability and unidirectionality hopefully offsets the additional time required. On the flip side, given its favorable properties in terms of recovering explanations it could be used to violate privacy which may be concerning from a social standpoint.

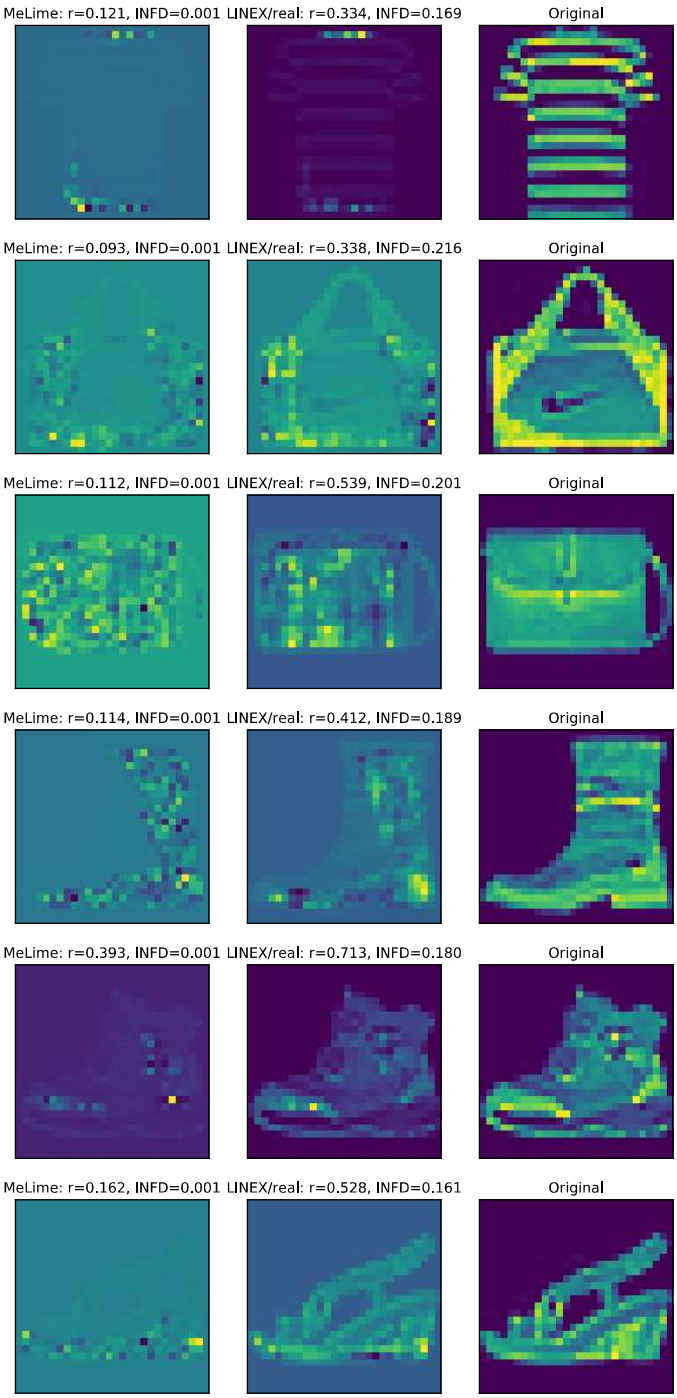


Figure 23: Error analysis for a chosen set of examples in FMNIST using MeLIME and LINEX/real methods. The three columns are the MeLIME feature attributions, LINEX/real feature attributions, and the original images. The rows correspond to different examples. We show the Pearson's correlation coefficient between feature attributions and mean of the original images from the respective classes (r) and instance-level infidelity (INFD) measures. LINEX seems to highlight important features like stripes in the t-shirt, handles of the bags, outlines of the boots/shoes more prominently, while MeLIME seems to overfit to the data while missing out on highlighting some key features prominently.

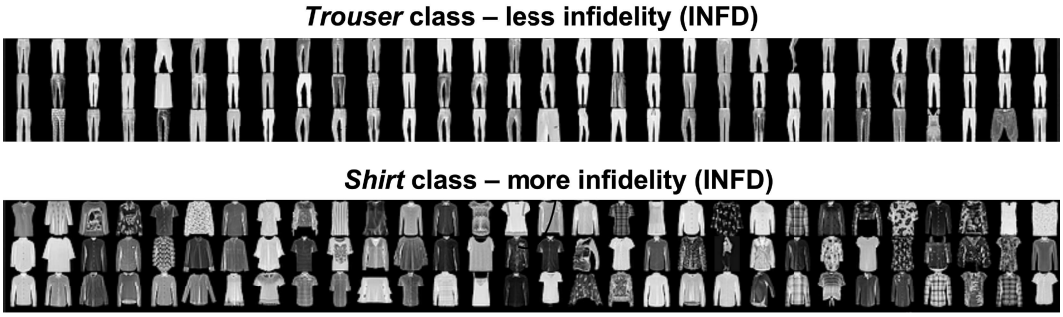


Figure 24: We see above that infidelity is lower for Trousers class for LINEX as compared with the Shirts class. A reason for this is that the trousers are more plain with less superfluous features such as the different designs in shirts. Since LINEX focuses on robust features focusing excessively on the designs is not critical for it to determine a shirt, albeit focusing on these designs might reduce infidelity. Advantage of it relying on robust features is however apparent when we look at other metrics such GI, CAC, CI and Υ as seen in Table 2 where it is much closer to or superior to MeLIME.

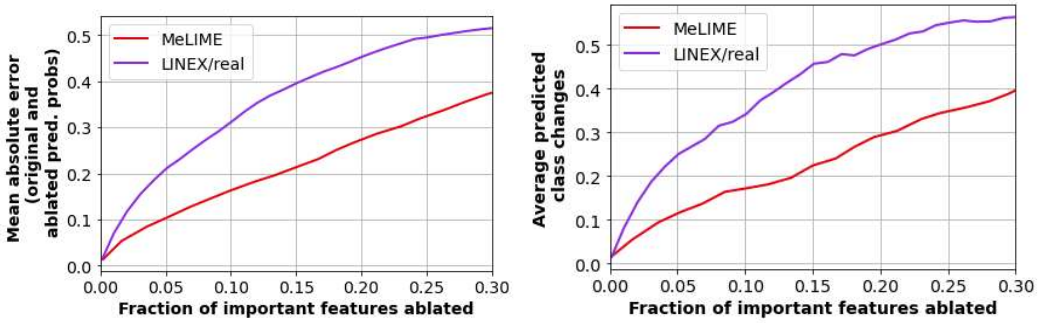


Figure 25: Ablation analysis to determine if the features deemed important by the explanation methods are actually considered important for prediction by the black box model. We see that features chosen by LINEX impact the prediction of the black box model much more than those chosen by MeLIME. This is true with respect to both MAE measure (left) between the predicted probabilities before and after ablation for winning (or argmax) class, and the change in predicted classes (right) before and after ablation. Higher values here mean that the features chosen by the explanations are more relevant for the black box to make its predictions. The maximum value of both measures is 1.0.

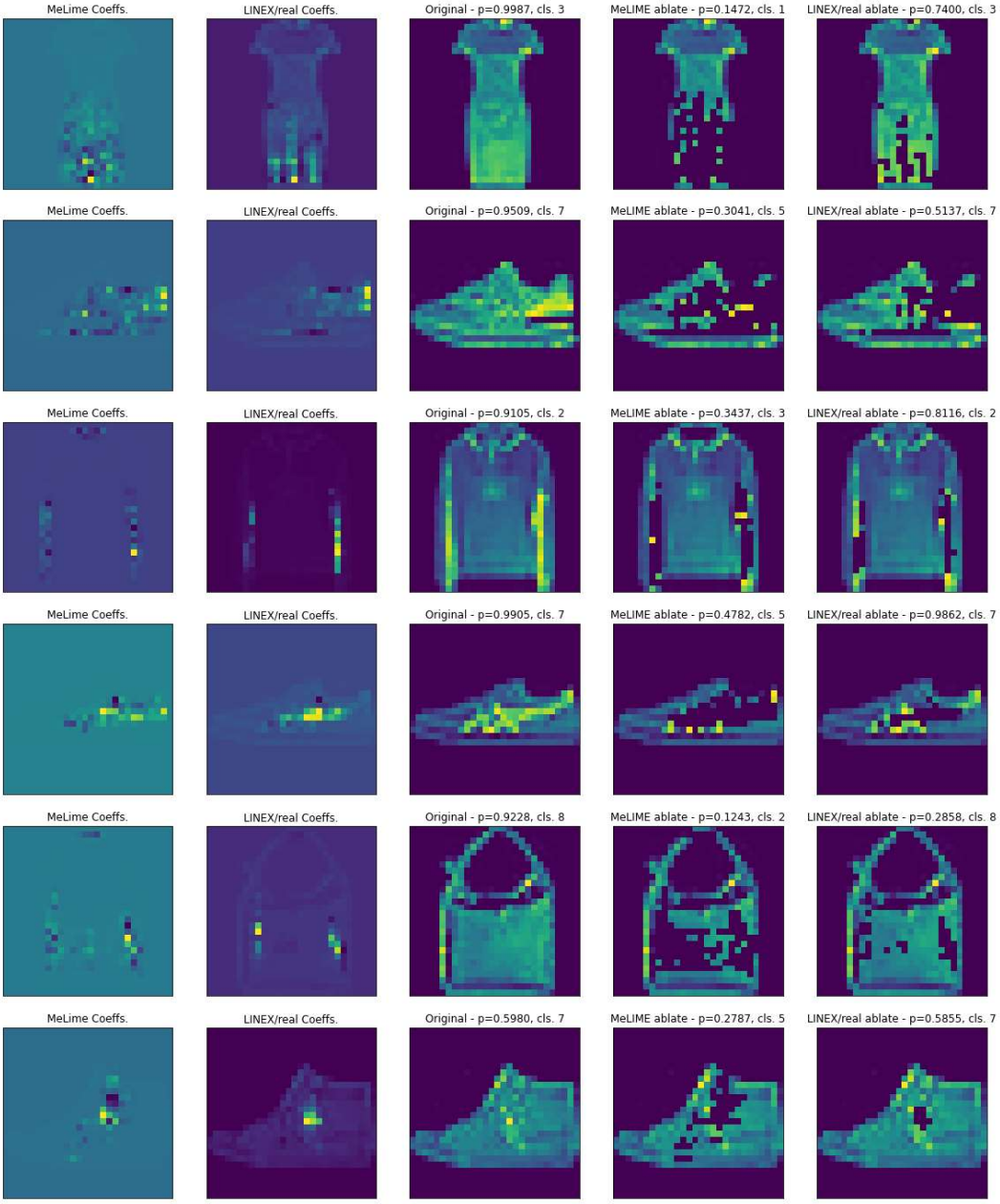


Figure 26: Error analysis for a chosen set of examples in FMNIST using MeLIME and LINEX/realistic methods, using ablation of important features. Each row shows results for a particular image. The columns show the: (a) MeLIME coefficients, (b) LINEX/realistic coefficients, (c) the original image along with its predicted class (cls.) and predicted probability for that class (p), (d) the image after MeLIME ablation along with the predicted probability for the original class (p) and the new class prediction (cls.), and (e) the image after LINEX/realistic ablation along with the predicted probability for the original class (p) and the new class prediction (cls.). The changes in prediction for MeLIME ablation for the six images are respectively from *Dress* to *Trouser*, *Sneaker* to *Sandal*, *Pullover* to *Dress*, *Sneaker* to *Sandal*, *Bag* to *Pullover*, and *Sneaker* to *Sandal*. No changes in classes are seen for LINEX ablation.

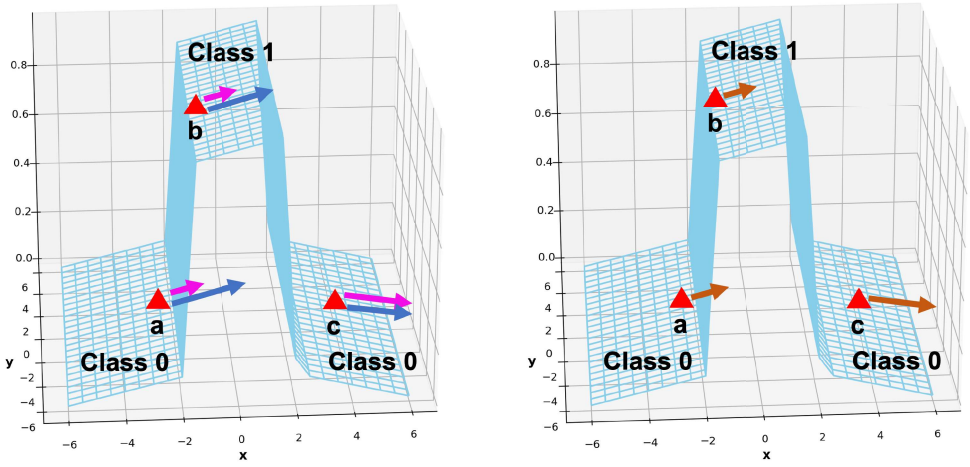


Figure 27: LIME (left) and LINEX (right) feature attributions for three points (a, b, c) for a synthetic data where we have Class 1 sandwiched between Class 0. For LIME, the different colors pink and blue correspond to feature attributions obtained with the small and large kernel width neighborhoods. Note how explanations for LIME change significantly (in magnitude) by kernel widths near the class boundaries, whereas the LINEX explanation remains stable, where it still picks up the important feature.

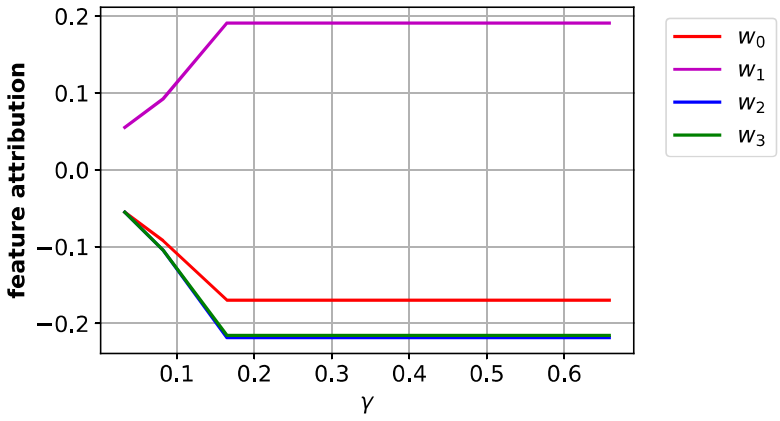


Figure 28: Feature attributions for the four features for an example in the IRIS dataset are shown above when varying γ. We used the same setting as in Section 5 for this experiment. The attributions increase smoothly as γ increases and stay constant after $\gamma \geq \min(|w_{1i}|, |w_{2i}|) \forall i$.

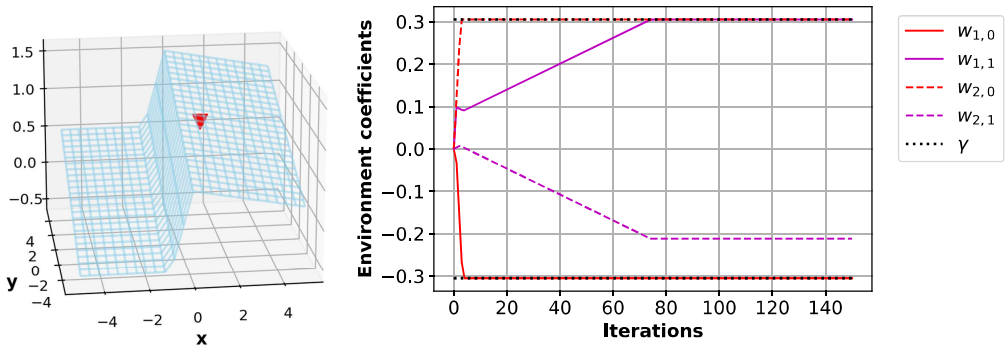


Figure 29: Left side: Explaining a scalar function in $\mathbb{R}^2$ at the point indicated by the triangle. The point is centered at $x = 1.0, y = 0.0$. The two environments are created by sampling multivariate normals with variances 0.5 and 2.0 respectively (samples not shown) centered at this point. Right side: Convergence of individual environment attributions. The attributions for first feature (x), $w_{1,0}$ and $w_{2,0}$, converge to γ and $-\gamma$ leading to the optimal attribution of 0. For the second feature (y) the optimal attribution ($w_{1,1} + w_{2,1}$) converges to a positive value.