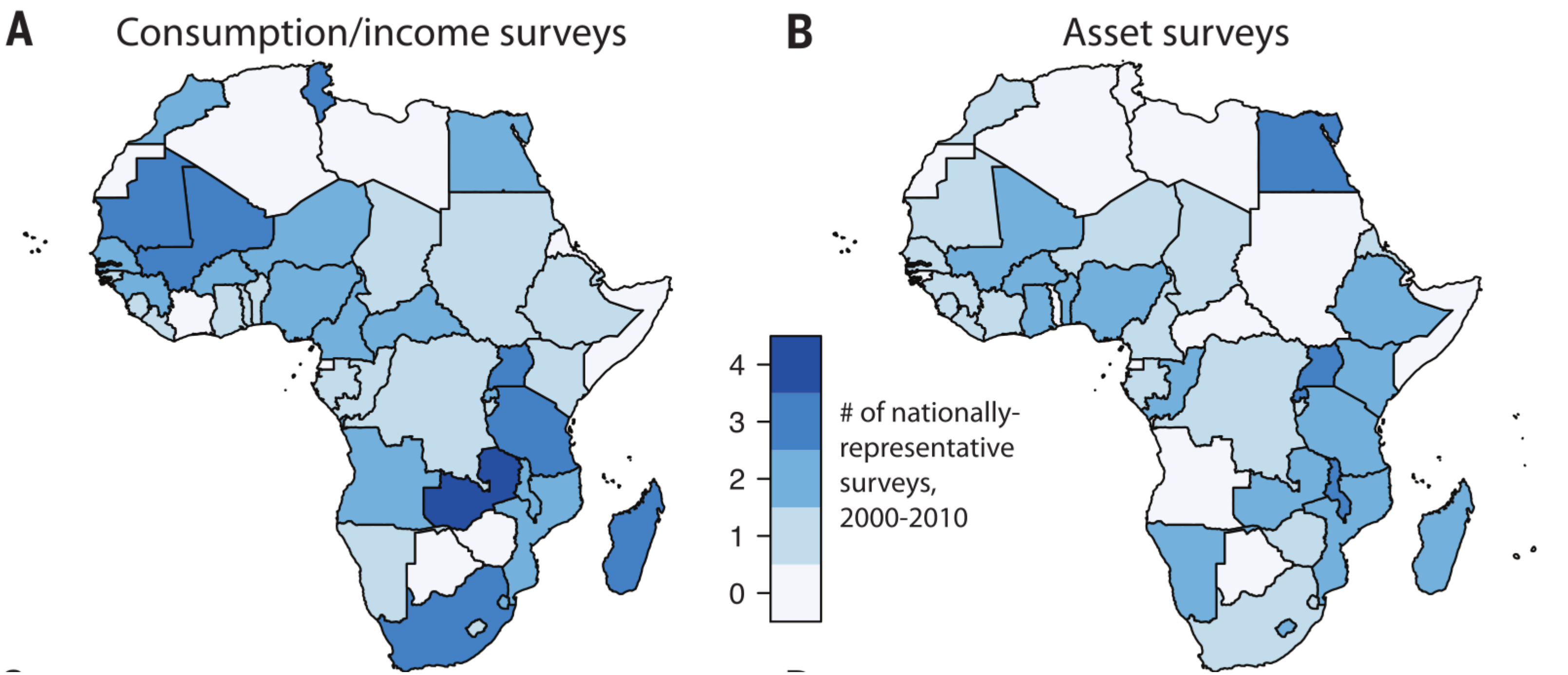
Introduction to Machine Learning

Lecture 18
Combining satellite imagery and machine learning to predict poverty

Goker Erdogan
26 – 30 November 2018
Pontificia Universidad Javeriana

Predicting poverty using satellite imagery

- Problem
  - Little data on poverty/wealth
    - Especially for African countries
  - Hard/expensive to collect this data


A
Consumption/income surveys
B
Asset surveys
4
3
2
1
0
# of nationally-representative surveys, 2000-2010

Predicting poverty using satellite imagery

- Problem
  - Little data on poverty/wealth
    - Especially for African countries
  - Hard/expensive to collect this data

- This paper
  - a machine learning approach for extracting socioeconomic data from
    - High-resolution daytime satellite imagery

Problem

- Two indicators of economic well-being
  1) Consumption expenditure
  2) Household asset score (asset wealth)

- Can we predict these from some other data?
  - Given some neighborhood (longitude/latitude), predict household consumption/asset wealth
  - Previous work used nightlight intensity

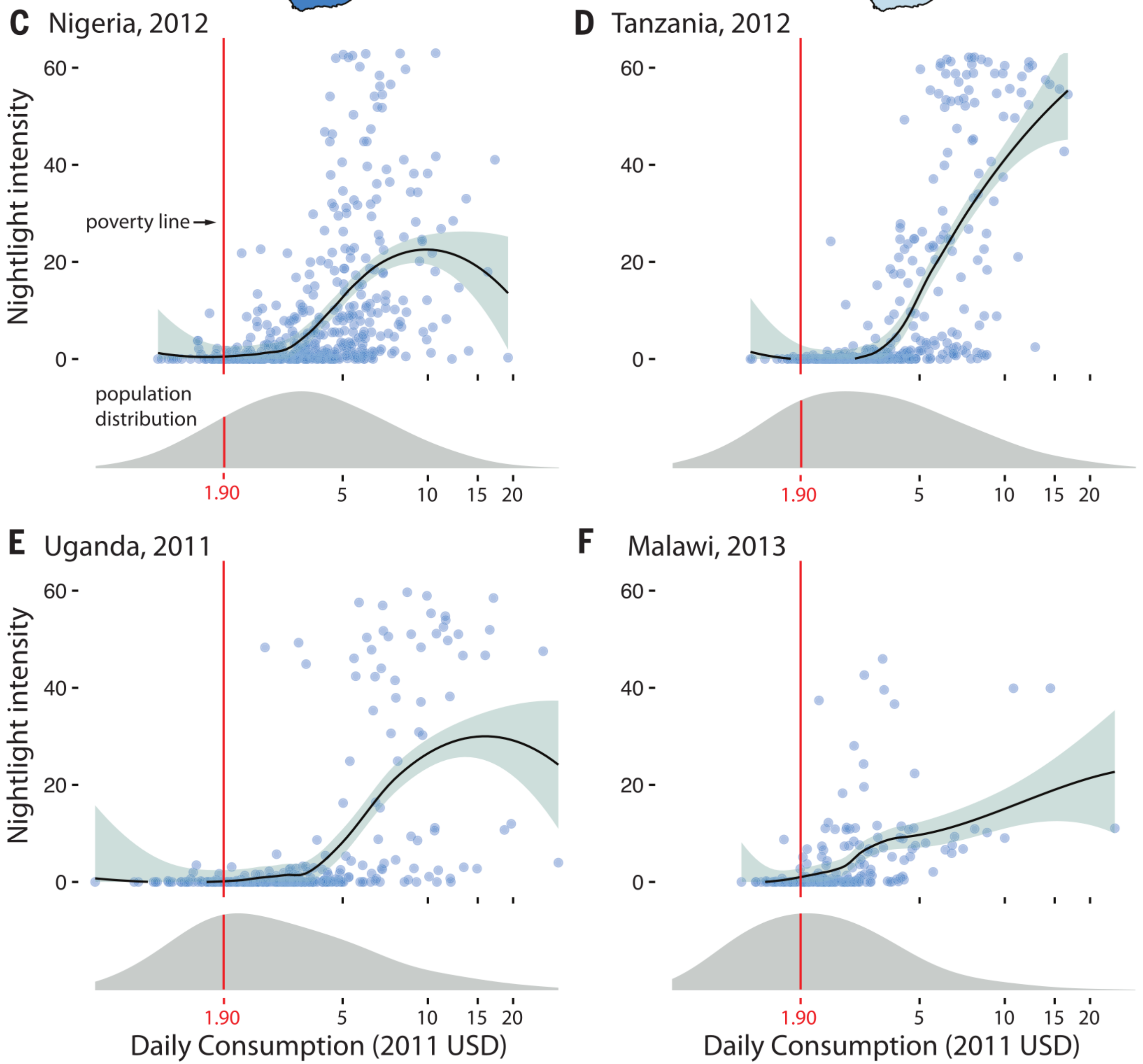


Fig. 1. Poverty data gaps. (**A**) Number of nationally representative consumption surveys occurring in each African country between 2000 and 2010. (**B**) Same as (A), for DHS surveys measuring assets. (**C** to **F**) Relationship between per capita consumption expenditure (measured in U.S. dollars) and nightlight intensity at the cluster level for four African countries, based on household surveys. Nationally representative share of households at each point in the consumption distribution is shown beneath each panel in gray. Vertical red lines show the official international extreme poverty line ($1.90 per person per day), and black lines are fits to the data with corresponding 95% confidence intervals in light blue.

Problem

- Two indicators of economic well-being
  1) Consumption expenditure
  2) Household asset score (asset wealth)

- Can we predict these from some other data?
  - Given some neighborhood (longitude/latitude), predict household consumption/asset wealth
  - Previous work used nightlight intensity
  - Works OK but not great

- Can we do better?
  - Use daytime satellite imagery

Approach

- Labeled data are scarce
  - Several hundred data points for each country
  - Difficult to train a large model

- Multi-step approach
  1) Train a large convolutional neural network on a data-rich proxy task
     - Nightlight estimation
  2) Extract features from satellite images using this network
  3) Train a linear regression model (with l_2 regularization) to predict consumption and asset wealth

Transfer learning

- Transfer learning
  - Training on a proxy task
  - Using the trained model for the task you are interested in

- Usually
  - Take a pretrained neural network
    - e.g., trained on ImageNet dataset to predict objects
  - Remove the last layer and add your output layer
  - Re-train the model (continuing from the pretrained weights)

Proxy task: nightlight estimation

- Start with 8-layer convolutional network VGG-F [2]
  - Trained on ImageNet dataset to recognize 1000 object classes

Arch.	conv1	conv2	conv3	conv4	conv5	full6	full7	full8
CNN-F	64x11x11 st. 4, pad 0 LRN, x2 pool	256x5x5 st. 1, pad 2 LRN, x2 pool	256x3x3 st. 1, pad 1 -	256x3x3 st. 1, pad 1 -	256x3x3 st. 1, pad 1 x2 pool	4096 drop- out	4096 drop- out	1000 soft- max

- Predict nighlight intensity
  - Formulate it as a classification problem
    - Three classes: low, medium, high
    - Found using mixture of Gaussians
  - Input daytime satellite image → output nightlight intensity

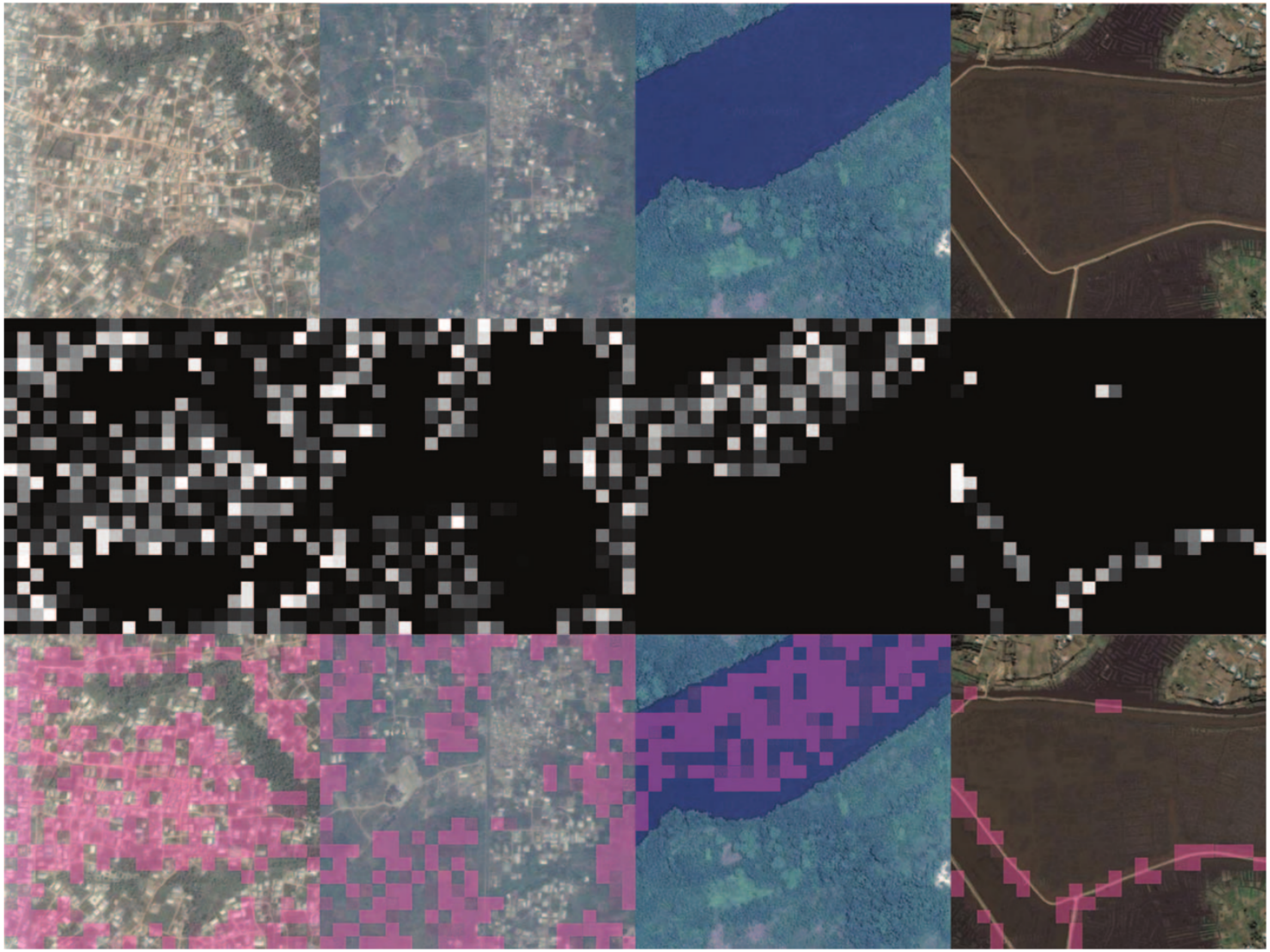


Fig. 2. Visualization of features. By column: Four different convolutional filters (which identify, from left to right, features corresponding to urban areas, nonurban areas, water, and roads) in the convolutional neural network model used for extracting features. Each filter "highlights" the parts of the image that activate it, shown in pink. By row: Original daytime satellite images from Google Static Maps, filter activation maps, and overlay of activation maps onto original images

Extracting features for poverty prediction

- Remove the output layer from convolutional network
- Use the outputs of full7 layer as your features for satellite images
  - A vector of 4096 values for an image

- For each cluster (neighborhood) in your data,
  - Take its satellite image
  - Feed into the convolutional network
  - Output is our image features for that cluster

- Predict consumption/asset wealth from these features
  - Much fewer features than using the images
  - Should work with little data

Predicting poverty from learned image features

- Use ridge regression to predict consumption/asset wealth from image features
  - Ridge regression: linear regression + l_2 regularization
  - Input: 4096 dim. image features
    - Pass satellite image through convolutional network
    - Take the output as your features
  - Output: consumption or asset wealth

- Do K-fold cross-validation to evaluate performance

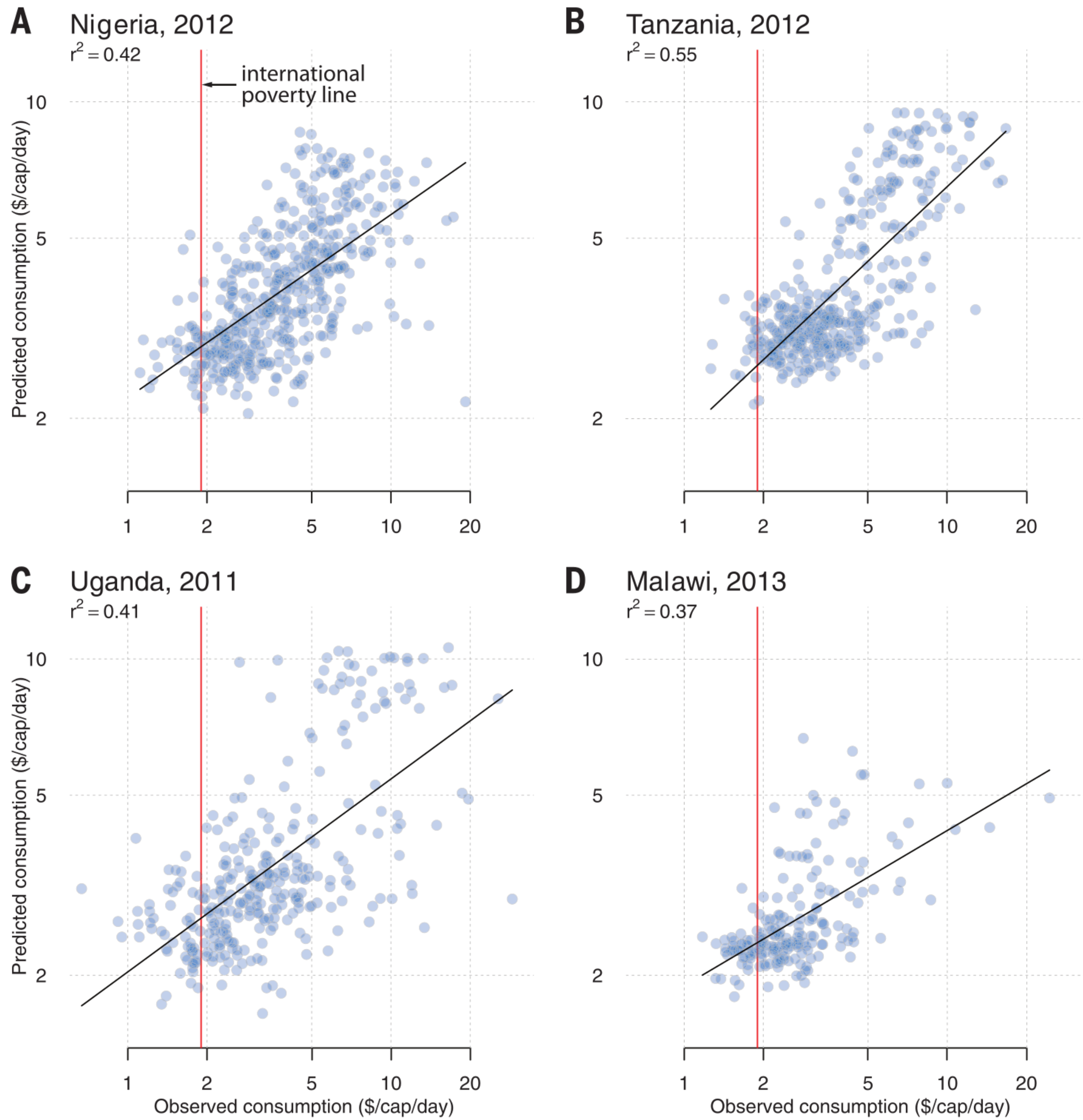


Fig. 3. Predicted cluster-level consumption from transfer learning approach (*y* axis) compared to survey-measured consumption (*x* axis). Results are shown for Nigeria (**A**), Tanzania (**B**), Uganda (**C**), and Malawi (**D**). Predictions and reported r^2 values in each panel are from fivefold cross-validation. Black line is the best fit line, and red line is international poverty line of $1.90 per person per day. Both axes are shown in logarithmic scale. Countries are ordered by population size.

Figure S3: **Predicted cluster-level asset index from transfer learning approach (y-axis) compared to DHS-measured asset index (x-axis) for 5 countries.** Predictions and reported r^2 values in each panel are from 5-fold cross validation. Both axes shown in log-scale. Black line is the best fit line.

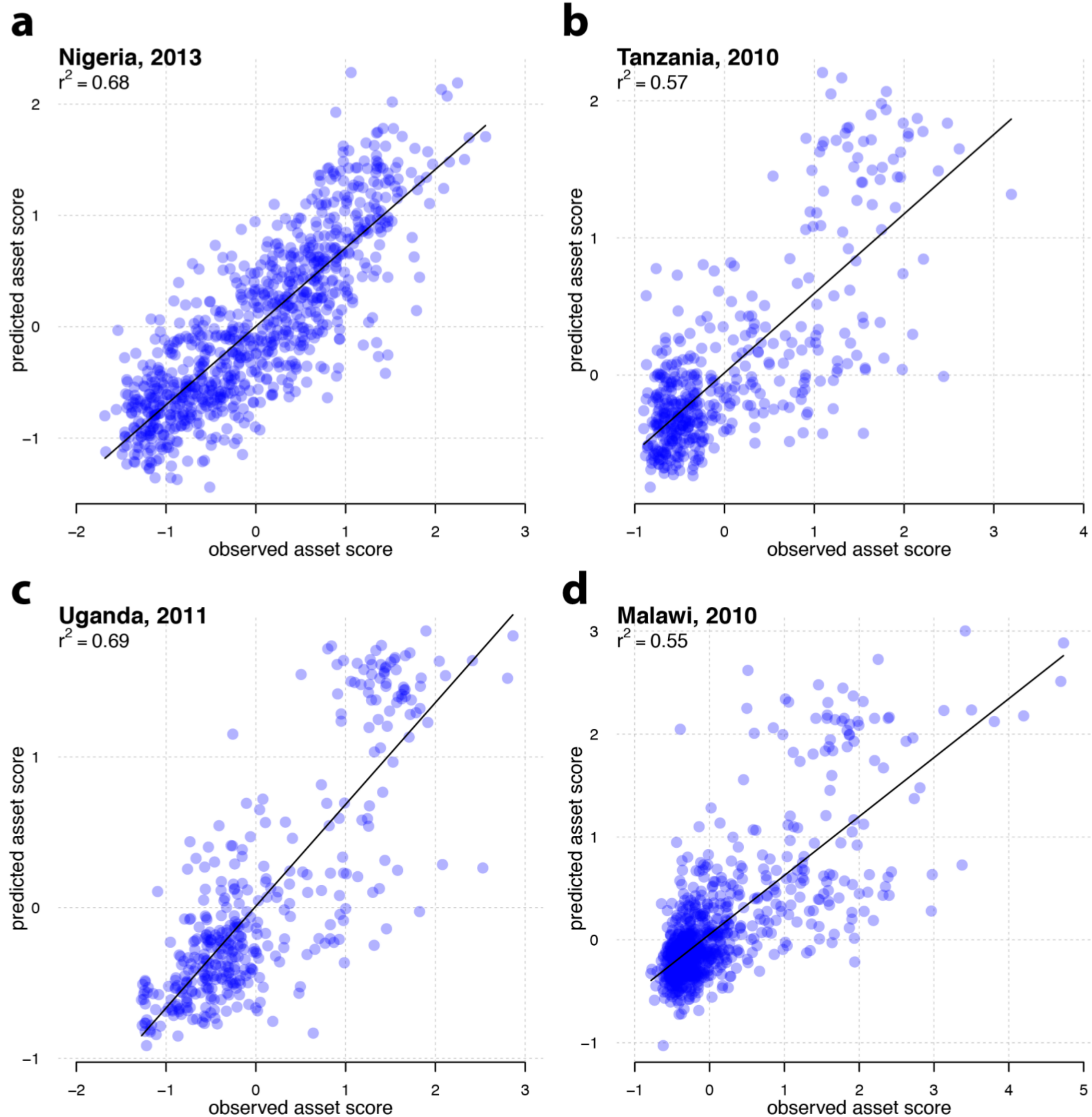
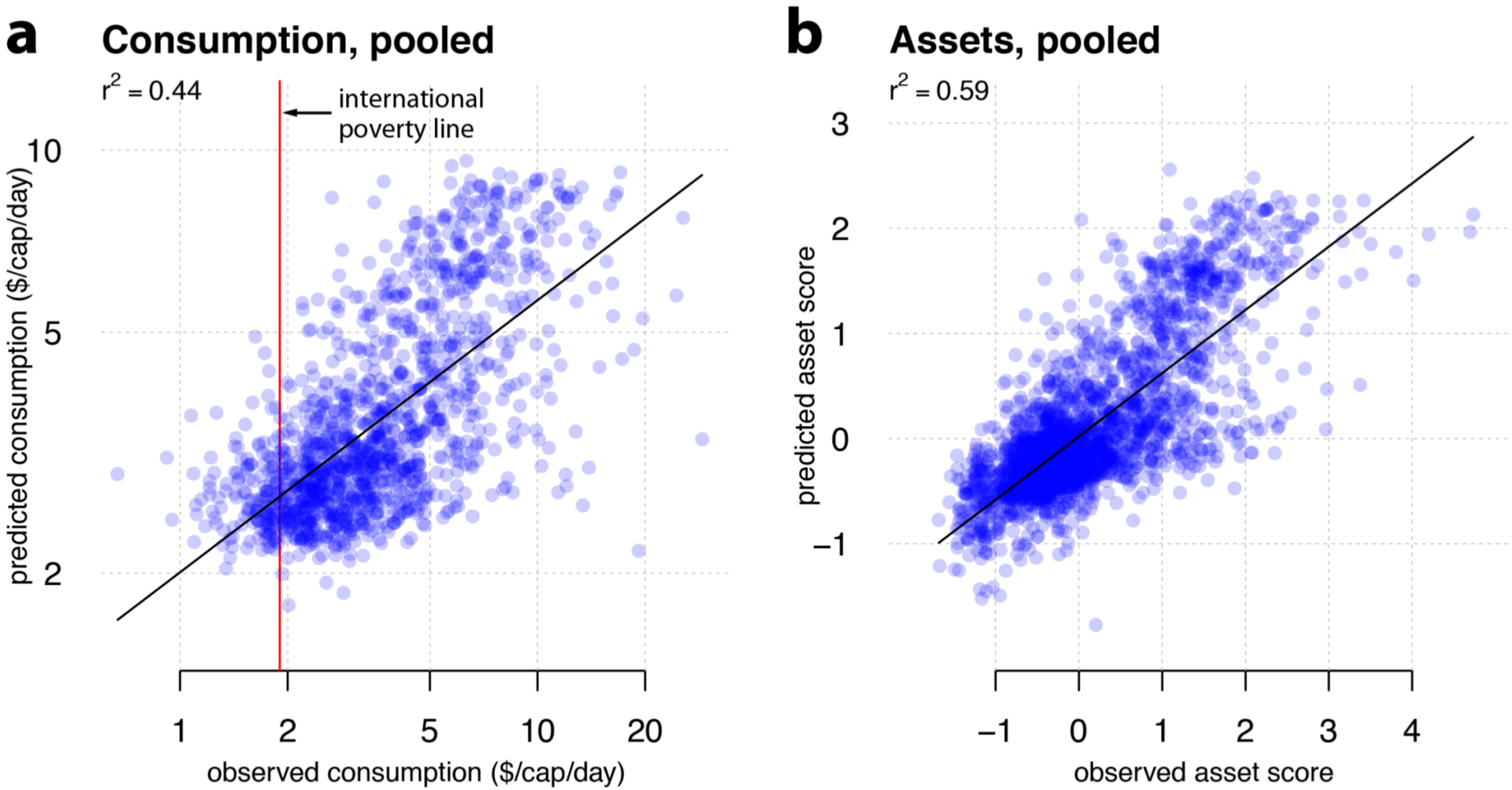


Figure S4: **Relationship between estimated and observed consumption (a) and assets (b)**, from a pooled model using data from all four LSMS countries (as in Figure 3) or all five DHS countries (as in Figure S4). Vertical red line in the left panel is the international poverty line (\$1.90 person^{-1} day^{-1}). Both axes shown in log-scale for consumption.



Comparison with prediction from nightlight intensity

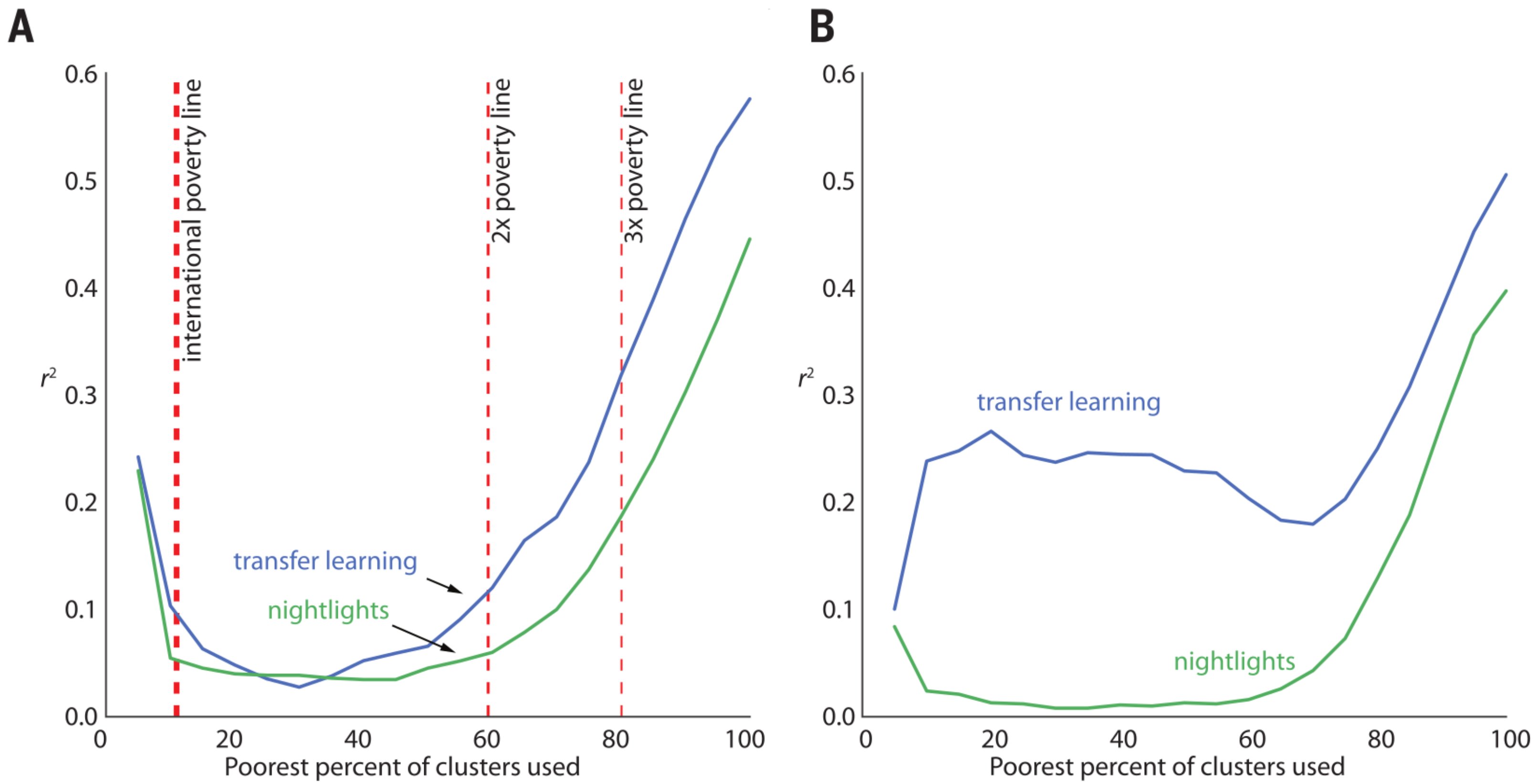


Fig. 4. Evaluation of model performance. (**A**) Performance of transfer learning model relative to nightlights for estimating consumption, using pooled observations across the four LSMS countries. Trials were run separately for increasing percentages of the available clusters (e.g., *x*-axis value of 40 indicates that all clusters below 40th percentile in consumption were included). Vertical red lines indicate various multiples of the international poverty line. Image features reduced to 100 dimensions using principal component analysis. (**B**) Same as (A), but for assets. (**C**) Comparison of r^2 of models trained on correctly assigned images in each country (vertical lines) to the distribution of r^2 values obtained from trials in which the model was trained on randomly shuffled images (1000 trials per country). (**D**) Same as (C), but for assets. Cross-validated r^2 values are reported in all panels.

Cross-border model generalization

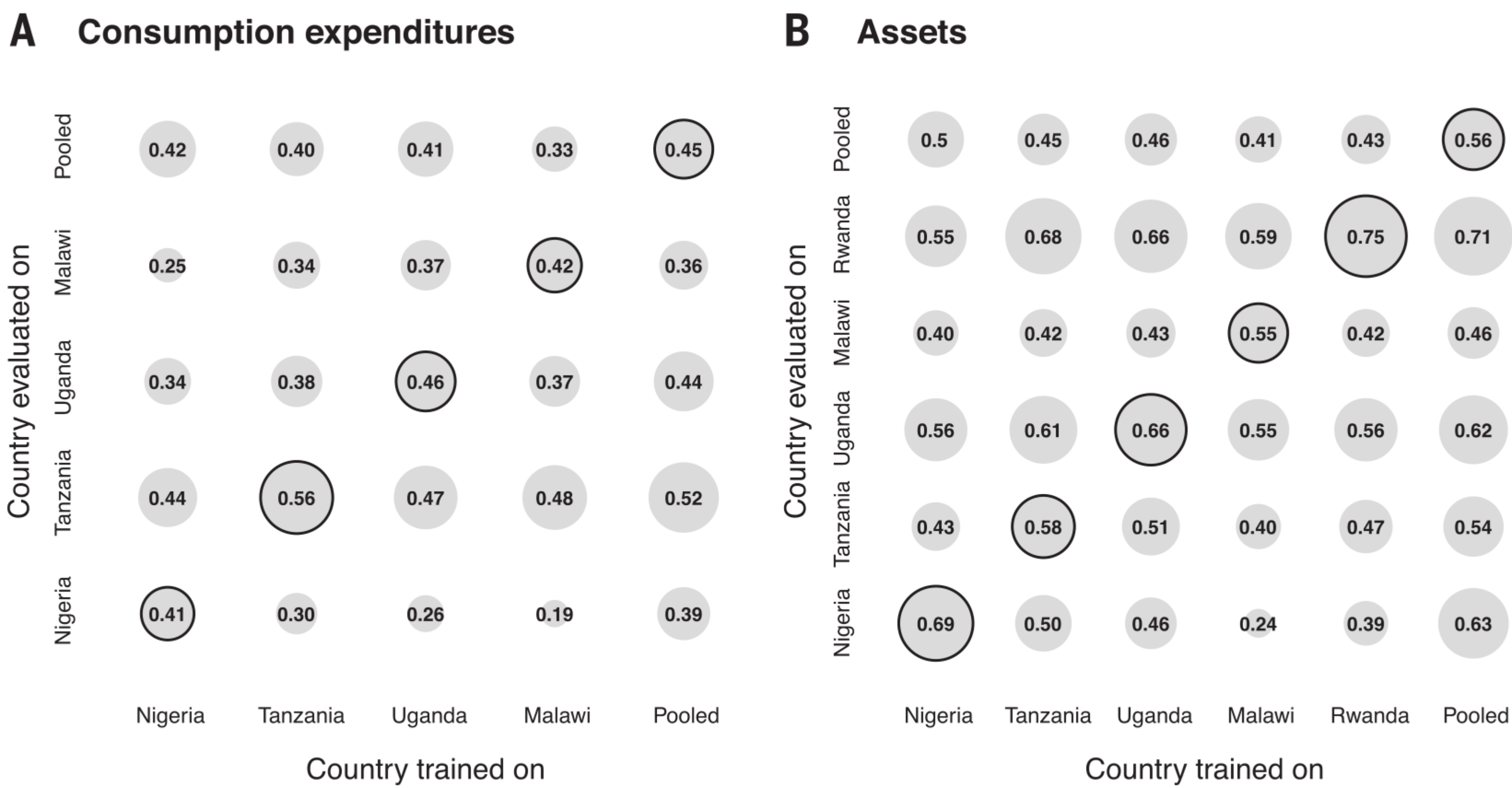
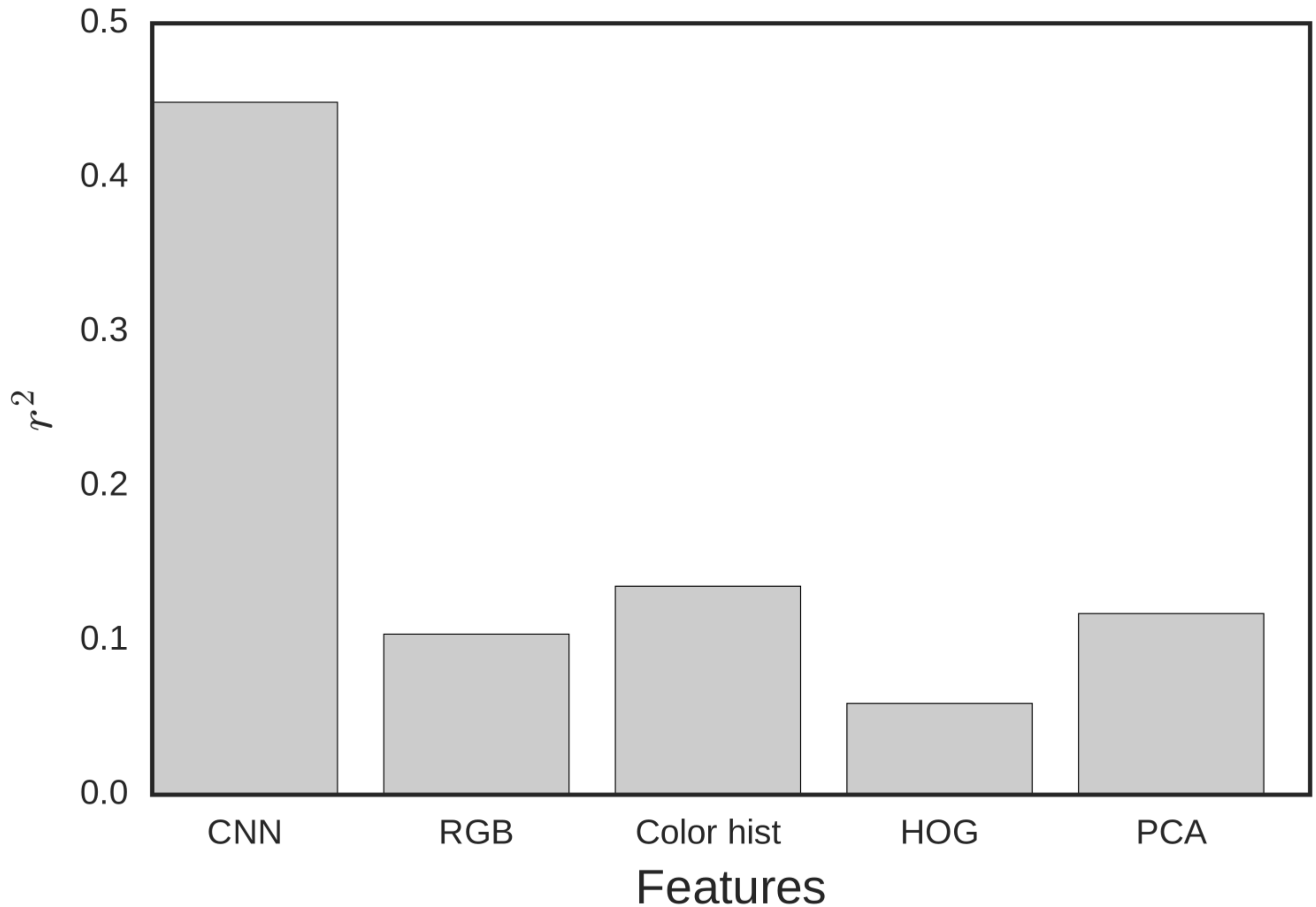


Fig. 5. Cross-border model generalization. (**A**) Cross-validated r^2 values for consumption predictions for models trained in one country and applied in other countries. Countries on *x* axis indicate where model was trained, countries on *y* axis where model was evaluated. Reported r^2 values are averaged over 100 folds (10 trials, 10 folds each). (**B**) Same as in (A), but for assets.

What if we used other image features?

Figure S8: **Comparison of CNN and alternative feature extraction methods.** Bar heights represent cross-validated r^2 achieved using five different approaches to feature extraction from daytime satellite imagery. See SM 2.3 for details.



Summary

- Predicting poverty from satellite images
  - Little data

- Transfer learning
  - Train on a proxy task to learn good features
  - Predict nighlight intensity from daytime satellite images

- Predict poverty using ridge regression from learned image features
  - Works better than predicting directly from nightlight intensity
  - Easy/cheap technique

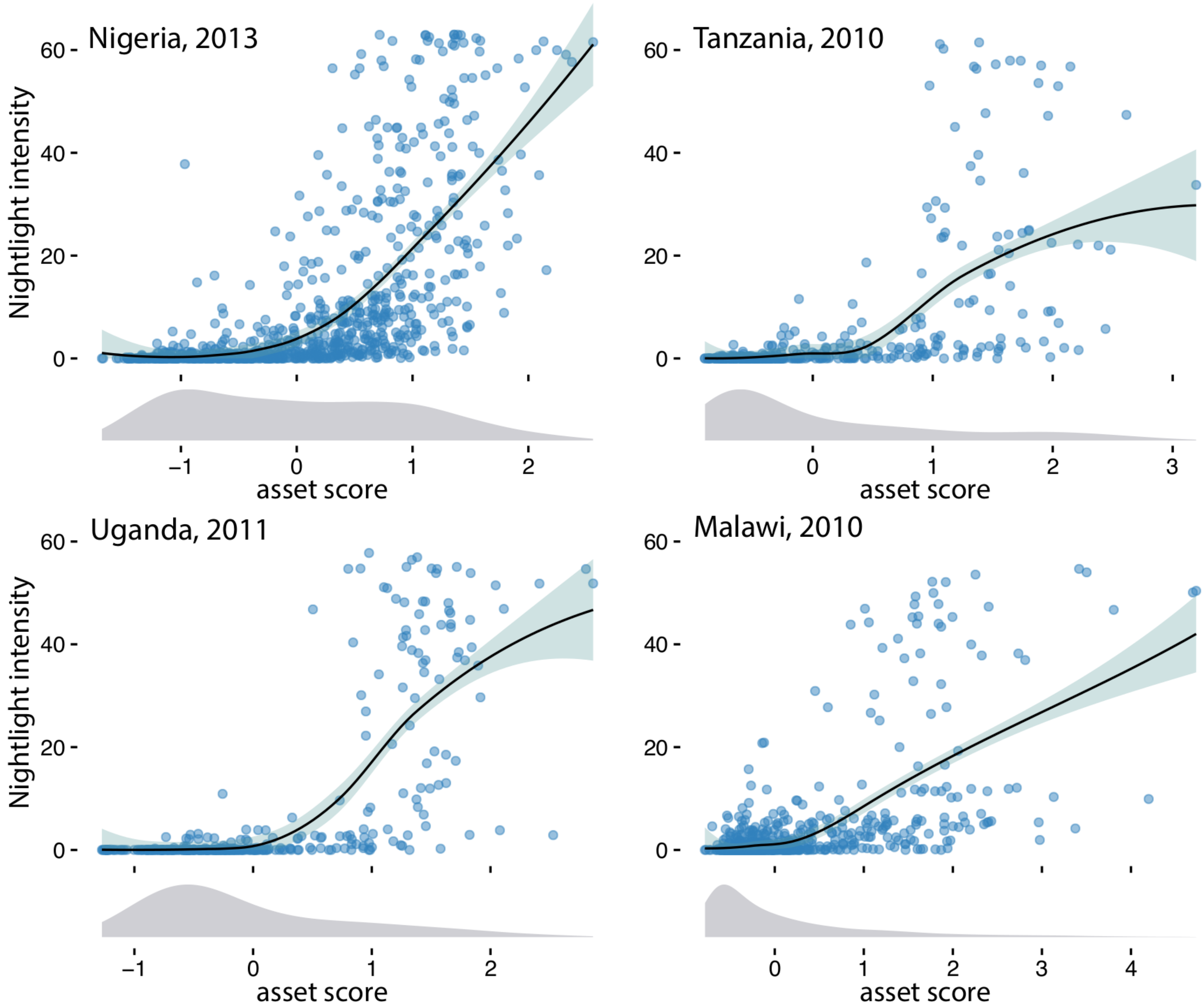
References

[1] Jean N., Burke M., Xie M., Davis M., Lobell D., Ermon S. Combining satellite imagery and machine learning to predict poverty. Science (2016).

[2] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the Devil in the Details: Delving Deep into Convolutional Networks, British Machine Vision Conference (2014).

backup

Figure S1: **Relationship between asset-based wealth index (from DHS) and nightlight intensity at the cluster level for five African countries.** Distribution of nationally-representative household-level wealth index scores shown beneath each panel in grey. Black lines are LOESS fits to the data with corresponding 95% confidence intervals in light blue.



Arch.	conv1	conv2	conv3	conv4	conv5	full6	full7	full8
CNN-F	64x11x11 st. 4, pad 0 LRN, x2 pool	256x5x5 st. 1, pad 2 LRN, x2 pool	256x3x3 st. 1, pad 1 -	256x3x3 st. 1, pad 1 -	256x3x3 st. 1, pad 1 x2 pool	4096 drop- out	4096 drop- out	1000 soft- max
CNN-M	96x7x7 st. 2, pad 0 LRN, x2 pool	256x5x5 st. 2, pad 1 LRN, x2 pool	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 x2 pool	4096 drop- out	4096 drop- out	1000 soft- max
CNN-S	96x7x7 st. 2, pad 0 LRN, x3 pool	256x5x5 st. 1, pad 1 x2 pool	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 x3 pool	4096 drop- out	4096 drop- out	1000 soft- max

TABLE 1

CNN architectures. Each architecture contains 5 convolutional layers (conv 1–5) and three fully-connected layers (full 1–3). The details of each of the convolutional layers are given in three sub-rows: the first specifies the number of convolution filters and their receptive field size as “num x size x size”; the second indicates the convolution stride (“st.”) and spatial padding (“pad”); the third indicates if Local Response Normalisation (LRN) [13] is applied, and the max-pooling downsampling factor. For full 1–3, we specify their dimensionality, which is the same for all three architectures. Full6 and full7 are regularised using dropout [13], while the last layer acts as a multi-way soft-max classifier. The activation function for all weight layers (except for full8) is the REctification Linear Unit (RELU) [13].