Analyzing Kepler Data Through Machine Learning: A Study of Neural Networks for Exoplanet Detection

Tyler Carlson[a)] and Renuka Rajapakse[b)]

Department of Physics, University of Massachusetts Dartmouth, Dartmouth, Massachusetts 02747, USA

[a)]Corresponding author: tcarlson1@umassd.edu

[b)]rrajapakse@umassd.edu

Abstract. As modern space observatories generate increasingly large photometric datasets, the need for scalable, automated tools to interpret transit signals has become critical. To address this scale problem, we examined which combinations of light-curve input representations are most informative and computationally efficient for a 1D convolutional neural network (CNN) to separate real transits from false-positive signals. Across eight systematically compared input configurations, local multiscale transit views achieved the best performance (validation AUC = 0.942, minimum validation loss = 0.30), outperforming global phase-curve inputs in validation AUC and minimum validation loss. Building on that result, we propose applying the trained CNN to unconfirmed KOI candidates to produce an independent, confidence-tiered ranking that augments NASA's existing candidate disposition-confidence framework. The same approach could later be extended to archival catalogs from the K2 mission (the repurposed continuation of Kepler) and future transit-survey candidate lists, improving the efficiency with which limited telescope time is allocated.

INTRODUCTION

Space missions such as the Kepler Space Telescope [1] and the Transiting Exoplanet Survey Satellite (TESS) monitor millions of stars simultaneously, producing photometric time series at a rate that far exceeds the capacity for manual scientific review. When a planet passes in front of its host star, it causes a brief, periodic dimming of the measured stellar flux (a transit signal) whose depth, duration, and morphology encode the planet's physical and orbital properties [2]. Kepler's automated pipeline flags these events as threshold crossing events (TCEs), a subset of which are promoted to Kepler objects of interest (KOIs). The KOI catalog, maintained by the NASA Exoplanet Science Institute, currently lists thousands of systems sorted into confirmed exoplanets, confirmed false positives, and unconfirmed candidates.

False positives, which can be attributed to signals from eclipsing binaries, instrumental artifacts, or stellar variability mimicking planetary transits, are the main challenge in this classification task [3, 4]. Follow-up spectroscopic observations required to confirm a candidate are both expensive and time limited; misallocating observatory resources to likely false positives carries a direct scientific cost. NASA maintains an existing automated vetting pipeline for candidate classification that reports Robovetter disposition scores. According to the DR25 catalog description, "the disposition score is simply the fraction of Monte Carlo iterations that result in a disposition of PC" [5], where PC denotes a *planetary candidate*. In this paper, we refer specifically to that Robovetter *disposition-confidence* score. The convolutional neural network (CNN) developed in the present work is ultimately intended to be applied to candidate systems to produce an independent confidence score, complementing NASA's Robovetter disposition-confidence score. The CNN could later be extended to archival candidate catalogs from the K2 mission (the repurposed continuation of the Kepler spacecraft after its reaction-wheel failures) and to future transit-survey candidate lists.

A major obstacle in this work is that raw Kepler light curves are heterogeneous, varying in length and cadence from star to star, and so must be reduced to a fixed-length numerical form before any CNN can ingest them. This is done by defining a *light-curve input configuration*: settings that include binning resolution, truncation of the light curve, and data smoothing. This work focuses on the first stage of that effort, identifying which configurations are most informative and computationally efficient. It is part of a larger project to train a high-accuracy supervised CNN on selected inputs and apply it to unconfirmed candidates to produce a prioritized, confidence-tiered observing list.

INPUT REPRESENTATION AND DATA PIPELINE

Dataset and Quality Control

Light-curve data were retrieved from the cumulative KOI table of the NASA Exoplanet Archive [6] using the `Lightkurve` Python package [7]. The raw catalog contained 2,746 confirmed exoplanets, 4,839 confirmed false positives, and 1,979 unconfirmed candidates. A quality-control pipeline filtered out entries showing corrupted files, insufficient data coverage, local flux anomalies within the transit window, secondary-eclipse signatures near phase 0.5, significant transit asymmetry, out-of-range transit durations, and flux noise exceeding signal-to-noise thresholds. Refining the dataset to keep only clean, intact systems improves neural network performance metrics (see Appendix A for full criteria).

Multiscale Light-Curve Representation

Every example of a given view must contain an identical number of data points across all stars, so the variable-length, variable-cadence Kepler light curves must be normalized by phase-folding and binning before training [2]. Using `Astropy` [8] for the underlying time-series and unit handling, each light curve was flattened to remove stellar variability trends, phase-folded at the known orbital period, and centered on the midtransit time T_0. From this, two classes of view were produced: a *global view* spanning the full phase profile (capturing out-of-transit stellar behavior) and a *local view* truncated to a window of $\pm 1.2 \times (T_{14}/2)$ around midtransit, capturing ingress, depth, and egress. Each view was resampled at multiple bin counts (200, 100, and 50 for local; 5,000 for global), and a Gaussian-convolved counterpart of each binned view was also created to provide a noise-suppressed representation of transit shape. Figure 2 in Appendix A illustrates these representations for a confirmed exoplanet, KIC 11241912.

Data Format: CSV to NPZ

The pipeline originally stored processed light curves as CSV files. This introduced substantial overhead: Python was forced to parse text, infer types, and convert to NumPy arrays on every training run, with floating-point precision loss. To eliminate this, all data were reprocessed and stored as compressed NumPy binary (`.npz`) archives. Each `.npz` file contained two `.npy` arrays per view, phase and flux stored as correlated arrays, and stellar metadata—Kepler ID, orbital period, transit epoch, transit duration, stellar radius, effective temperature, and surface gravity. This format is natively compatible with TensorFlow [9], loads in a single function call, preserves full floating-point precision, and allows for training without any needed conversion. Metadata embedded in each file will serve as further filtering and feature inputs in future model iterations.

Input Comparison Results

Eight CNN configurations were trained under identical hyperparameters, differing only in which combination of views was supplied as input. Figure 1 summarizes training AUC, validation AUC, training accuracy, validation accuracy, minimum validation loss, and best epoch for each variant. Here AUC denotes the area under the receiver operating characteristic curve, a threshold-independent measure of how reliably the network's confidence ranking separates genuine transits from false positives (1.0 being perfect separation and 0.5 no better than chance). The strongest overall configuration is Local Data, which loads all local transit views simultaneously and ranks best on both validation AUC and minimum validation loss. This supports the value of multiscale local binning: lower-bin views capture the broad transit profile, while higher-bin views preserve finer characteristics such as ingress and egress shape.

By contrast, Global Data performs weakest across most validation metrics, showing that the current network does not use global phase information well on its own. Some false-positive types can contain useful out-of-transit signatures, but extracting them likely requires more targeted preprocessing, filtering, and pooling; global inputs thus appear most valuable when merged with local branches rather than used in isolation.

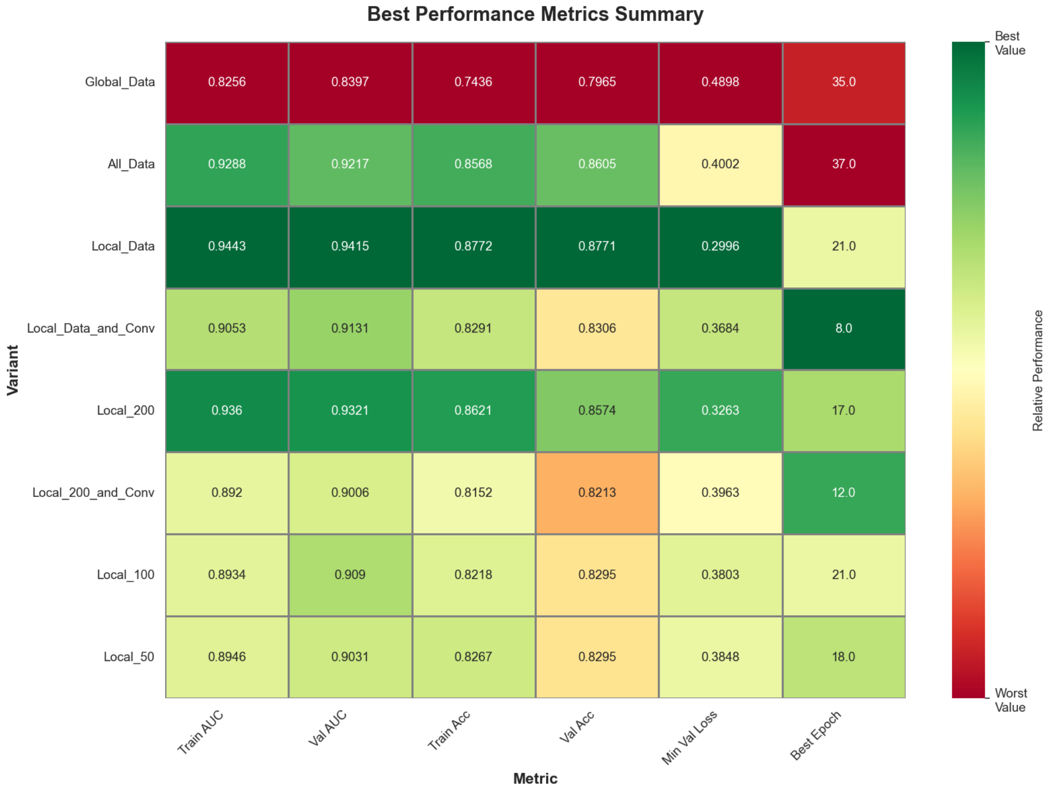

FIGURE 1. Best-performance summary for all eight CNN input configurations. Cells show raw metric values, including best epoch as a raw epoch count. Color reflects relative performance within each metric column. The worst value in a column maps to red and the best to green. For minimum validation loss and best epoch, lower values are treated as better in the color mapping. Local Data performs best overall, whereas Global Data performs worst across most validation metrics.

SUPERVISED CNN TRAINING AND PERFORMANCE

The best-performing configuration (Local Data, all three local scales without convolved counterparts) trained the final supervised model. The parallel 1D CNN processes each view through an independent convolution–pooling branch before hierarchical merging into fully connected layers and a sigmoid output node. This produces a planet-likeness score $p \in [0,1]$ whose higher values indicate closer resemblance to a genuine exoplanet transit (see Appendix B). All models were trained on GPU hardware using TensorFlow with binary cross-entropy loss, an Adam optimizer at base learning rate 10^{-4} with dynamic adjustment, early stopping at patience 15 over 100 epochs, and class weights of 1.64 (exoplanet) and 0.72 (false positive) to compensate for class imbalance. Architecture details are given in Appendix B; full training curves are shown in Appendix C.

The current supervised model achieves a validation AUC of 0.942 and a minimum validation loss of 0.30, a strong baseline for a first-pass vetting classifier. Several pathways could improve these metrics further: augmenting the training pool with NASA's catalog of *injected* simulated transits [10], refining the data-cleaning thresholds to reduce residual label noise, and tuning the convolutional filter widths and pooling strides, particularly on the global branch. These directions are detailed in Appendix C.

INFERENCE ON UNCONFIRMED CANDIDATES AND PRIORITIZATION

The final stage of this project will apply the trained supervised network to the 1,851 unconfirmed KOI candidates. Once the model demonstrates sufficient validation performance (including high AUC, stable precision–recall, and acceptable calibration on a held-out test set), it can be applied to the unlabeled candidates. Because the network has learned to separate planetary from nonplanetary light-curve shapes, its output score p provides a planet-likeness ranking rather than a calibrated probability; with further calibration, it could later be interpreted as an estimated probability.

Candidates will be sorted in descending order of this score and organized into three confidence tiers: **Tier 1** (high confidence, p above a precision-optimized threshold) — systems whose transit shape most closely resembles confirmed exoplanets, recommended for immediate spectroscopic or radial velocity follow-up; **Tier 2** (intermediate confidence) — systems warranting additional statistical vetting such as centroid shift analysis or transit timing variation searches before allocating telescope time; and **Tier 3** (low confidence) — systems whose signals most closely resemble known false-positive shapes. After sorting, each entry's Kepler ID and stored stellar metadata will be reattached so the output is a fully annotated, observatory-ready prioritized list.

Although NASA provides Robovetter disposition scores, the proposed pipeline will produce an independent CNN-derived ranking based only on learned light-curve characteristics. This ranking will not replace existing vetting products, but it will function as a practical prioritization metric. Candidates with high CNN scores can be inspected first, while low-scoring systems can be deprioritized or flagged for additional diagnostic checks before telescope time is allocated.

CONCLUSION

This paper aims to establish how light-curve data should be represented for machine-learning-assisted exoplanet candidate vetting. We find that multiscale local transit views, stored in compressed `.npz` format, provide the most efficient and informative input for a 1D CNN classifier, achieving a validation AUC of 0.942 on the Kepler KOI catalog. This input-representation study is the first stage of a larger project that will progress through supervised classification and inference on unconfirmed candidates toward a confidence-tiered candidate list designed to complement NASA's Robovetter-based vetting outputs with a distinct CNN confidence value. With continued improvements to data cleaning, training-pool augmentation, and hyperparameter optimization, the broader pipeline could, after further validation, grow into a scalable vetting tool for Kepler and future transit surveys such as TESS.

ACKNOWLEDGMENTS

We wish to acknowledge Dr. Sarah Caudill of the University of Massachusetts Dartmouth Department of Physics for her guidance and support. We would also like to thank Mr. Mark Munkacsy for his advice and feedback. This work was supported in part by the U.S. Office of Naval Research and the Massachusetts Space Grant Consortium (NASA MSGC).

REFERENCES

1. W. J. Borucki, D. Koch, G. Basri, N. Batalha, T. Brown, D. Caldwell, J. Caldwell, J. Christensen-Dalsgaard, W. D. Cochran, E. DeVore, et al., Science **327**, 977 (2010).
2. C. J. Shallue and A. Vanderburg, Astron. J. **155**, 94 (2018).
3. T. D. Morton, Astrophys. J. **761**, 6 (2012).
4. C. J. Burke and J. Catanzarite, "Planet detection metrics: Per-target detection contours for data release 25," Tech. Rep. KSCI-19109-002 (NASA Ames Research Center, Moffett Field, CA, 2017).
5. S. E. Thompson, J. L. Coughlin, K. Hoffman, F. Mullally, J. L. Christiansen, C. J. Burke, S. Bryson, N. Batalha, M. R. Haas, J. Catanzarite, et al., Astrophys. J. Suppl. Ser. **235**, 38 (2018).
6. R. L. Akeson, X. Chen, D. Ciardi, M. Crane, J. Good, M. Harbut, E. Jackson, S. R. Kane, A. C. Laity, S. Leifer, et al., Publ. Astron. Soc. Pac. **125**, 989 (2013).
7. Lightkurve Collaboration, "Lightkurve: Kepler and TESS time series analysis in Python," Astrophysics Source Code Library, ascl:1812.013 (2018).
8. Astropy Collaboration, T. P. Robitaille, E. J. Tollerud, P. Greenfield, M. Droettboom, E. Bray, T. Aldcroft, M. Davis, A. Ginsburg, A. M. Price-Whelan, et al., Astron. Astrophys. **558**, A33 (2013).
9. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., "TensorFlow: Large-scale machine learning on heterogeneous systems" (2015), software available from https://www.tensorflow.org.
10. J. L. Christiansen, "Planet detection metrics: Pixel-level transit injection tests of pipeline detection efficiency for data release 25," Tech. Rep. KSCI-19110-001 (NASA Ames Research Center, Moffett Field, CA, 2017).

Appendix A

Quality-control filters. Entries were rejected for any of the following: corrupted or unreadable files; fewer than a minimum number of in-transit cadences after folding; local flux maxima detected within the transit window (indicating systematics or an anomalous in-transit feature); transit asymmetry beyond a set tolerance; a secondary-eclipse signature near phase 0.5; transit duration outside physically motivated bounds; out-of-transit flux standard deviation exceeding a noise-floor threshold; or a transit signal-to-noise ratio below the detection significance threshold. Starting from 2,746 confirmed exoplanets and 4,839 confirmed false positives, these filters yielded 1,969 and 4,482 retained examples, respectively.

View construction. After flattening and phase-folding, each system was processed into the following views, stored as correlated phase and flux `.npy` arrays within a single `.npz` archive:

- Global unbinned (raw cadence) and global 5,000-bin — full orbital phase, capturing out-of-transit stellar behavior.
- Local 200-bin, 100-bin, and 50-bin — truncated to $\pm 1.2 \times (T_{14}/2)$ around midtransit; the three resolutions capture ingress/egress detail at 200 bins, overall transit shape at 100 bins, and slope and depth at 50 bins.
- Gaussian-convolved counterpart of each binned view (kernel width scaled to bin size) — noise-suppressed representation emphasizing transit envelope morphology.

Appendix B

The network uses a multi-input parallel 1D CNN in which each view is processed through an independent branch before merging. Each branch applies convolutional filters along the time axis to learn local temporal features, ingress slope, flat-bottom depth, and egress curvature, followed by max-pooling to reduce dimensionality while retaining the most salient activations. This is repeated across two convolutional depths before flattening. When convolved counterparts are included, the raw and convolved branches for the same scale are concatenated into a pair-merged vector.

Local-scale pair-merged (or individual) vectors are concatenated into a combined local representation, which is then passed through a small dense layer and dropout for regularization. If a global branch is present, it is processed separately and merged with the combined local vector at a second concatenation stage. The final merged representation passes through two fully connected dense layers with dropout before a single sigmoid-activated output node producing the classification score p.

All eight variants share the same branching and merging logic; only the set of active input branches changes. This ensures that observed performance differences are attributable solely to the information content of each input combination. All models were trained on GPU hardware using TensorFlow with the Adam optimizer, binary cross-entropy loss, class weights (1.64 for exoplanet, 0.72 for false positive), a base learning rate of 10^{-4} with dynamic adjustment, and early stopping at patience 15 over a maximum of 100 epochs. Validation and test sets each constituted 15% of labeled examples, stratified by class.

Appendix C

Figure 3 shows training loss, validation loss, training accuracy, validation accuracy, training AUC, and validation AUC across all epochs for every variant. Several patterns are noteworthy. The Global Data model exhibits high, slowly decaying validation loss (> 1.0 throughout) and volatile validation accuracy, indicating that the high-dimensional global array is difficult to optimize when used in isolation. Some false-positive classes likely contain informative out-of-transit structure, but those cues appear weak without stronger preprocessing and pooling. The All Data model shows an extreme training loss spike (> 30) at initialization followed by rapid decay, a consequence of large gradient norms when the global and local branches are simultaneously active without relative weighting. All local-scale models converge smoothly within 10–17 epochs, with training and validation AUC tracking closely, indicating minimal overfitting in the current configuration.

Three concrete pathways for improving the supervised model's performance are identified:

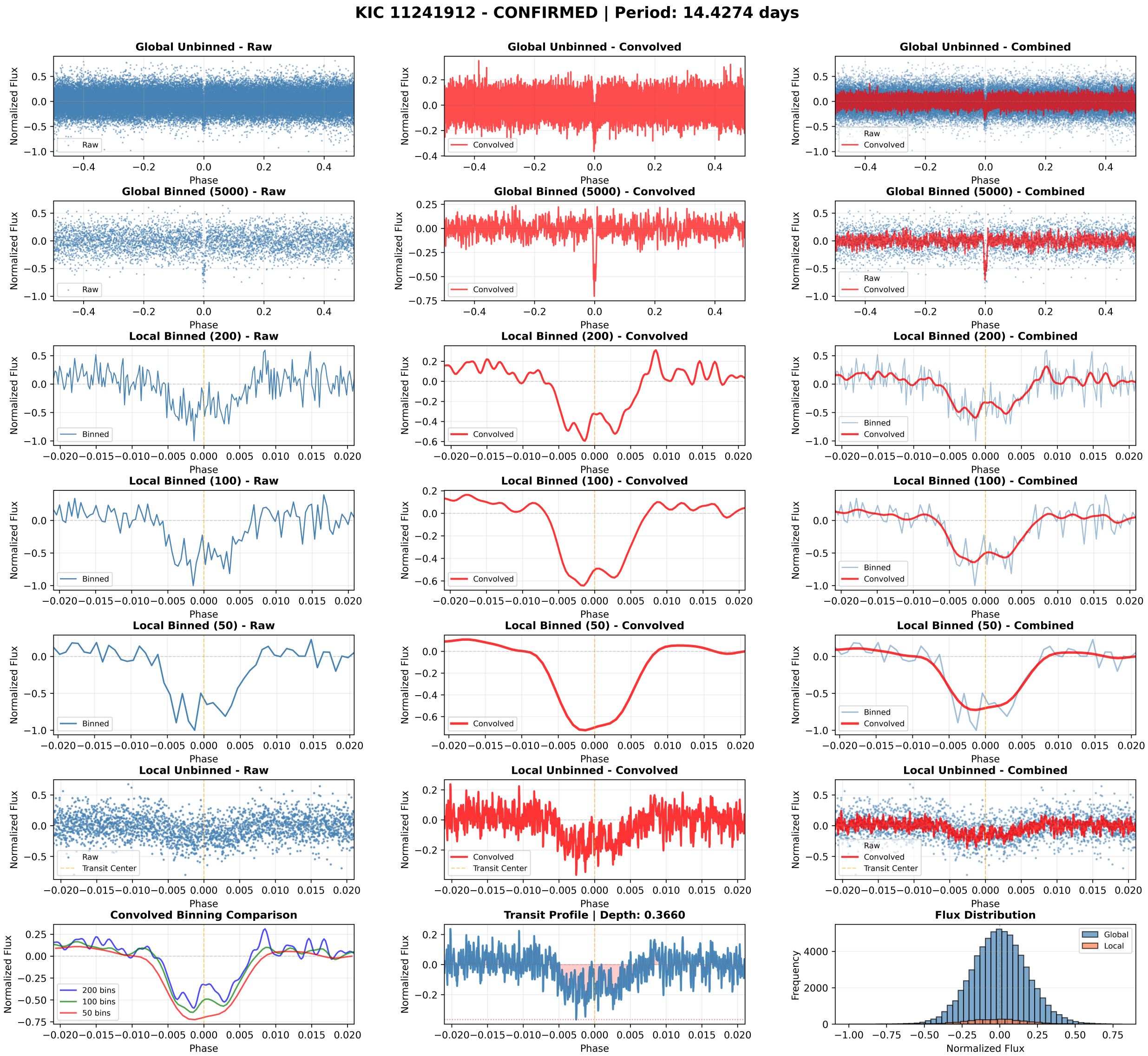

FIGURE 2. Multiscale light-curve representations for KIC 11241912 (confirmed exoplanet, period 14.4274 d, transit depth 0.366 in normalized flux). *Left column:* raw binned arrays. *Center:* Gaussian-convolved counterparts. *Right:* overlay of both. Rows correspond to the global unbinned, global 5,000-bin, local 200-bin, local 100-bin, local 50-bin, and local unbinned views. The number in parentheses beside each label indicates the number of bins used for that representation. The yellow dashed line in each plot marks the transit center. Bottom row shows the convolved binning comparison at all three local resolutions, the in-transit profile with measured depth, and the normalized flux distribution.

Injected signal augmentation. NASA publishes catalogs of artificially injected transit signals added to real Kepler light curves at known parameters [10]. Incorporating these into the training pool would substantially increase labeled example diversity and total count, directly improving generalization to weak or ambiguous transit morphologies.

Data cleaning refinement. The current quality-control pipeline applies fixed thresholds. Adaptive or percentile-based rejection criteria, combined with outlier clipping within individual transits, would reduce label noise from borderline systems that passed the current filters despite marginal data quality.

Hyperparameter and architecture tuning. The convolutional filter widths and pooling strides applied to the global branch are the most likely avenue for improving its contribution to classification. Prior work [2] found that wide filters on the global view (capturing multitransit periodicity rather than single-transit morphology) substantially improved global branch utility. A systematic grid search over filter size, number of convolutional layers per branch,

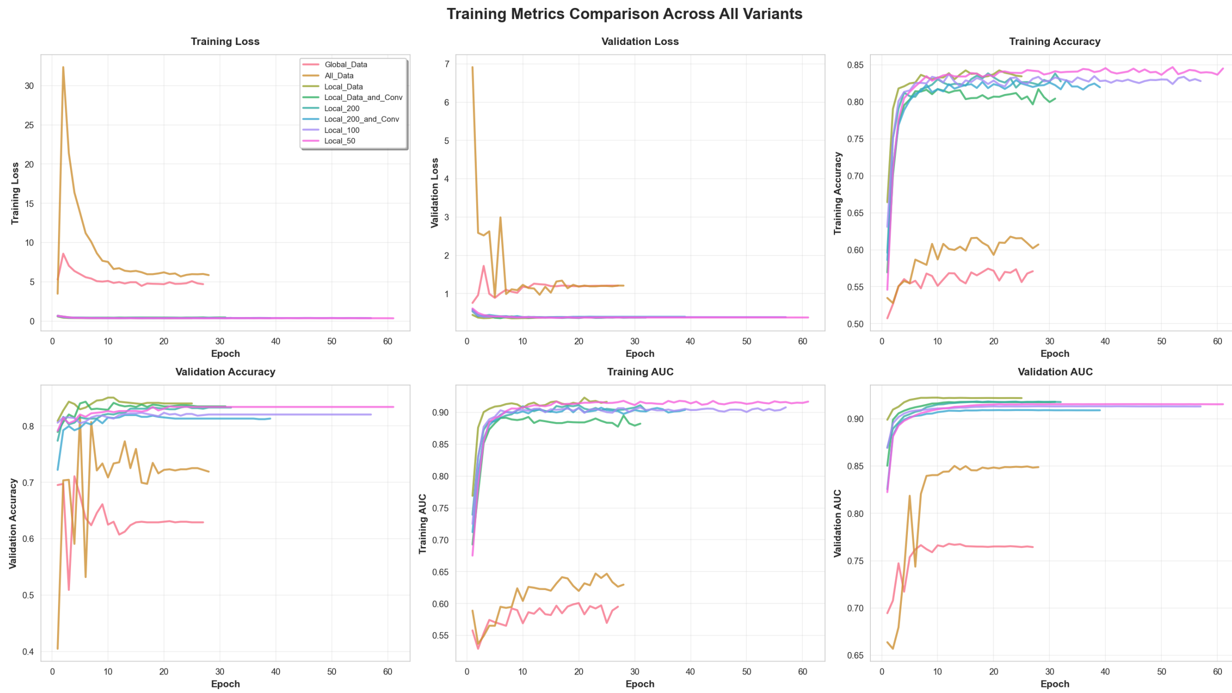

FIGURE 3. Training and validation loss, accuracy, and AUC across all epochs for all eight CNN input variants. Local-scale models (green, teal, blue, purple, pink) converge rapidly and stably. Global-Only (salmon) and All-Data (orange) models have high loss and rough validation curves.

and dense layer width would likely yield additional AUC gains across all configurations. In addition, training with curated false-positive subclasses (or subtype-aware loss weighting) may help the global branch learn rare out-of-transit indicators and contribute more reliably when merged with local branches.