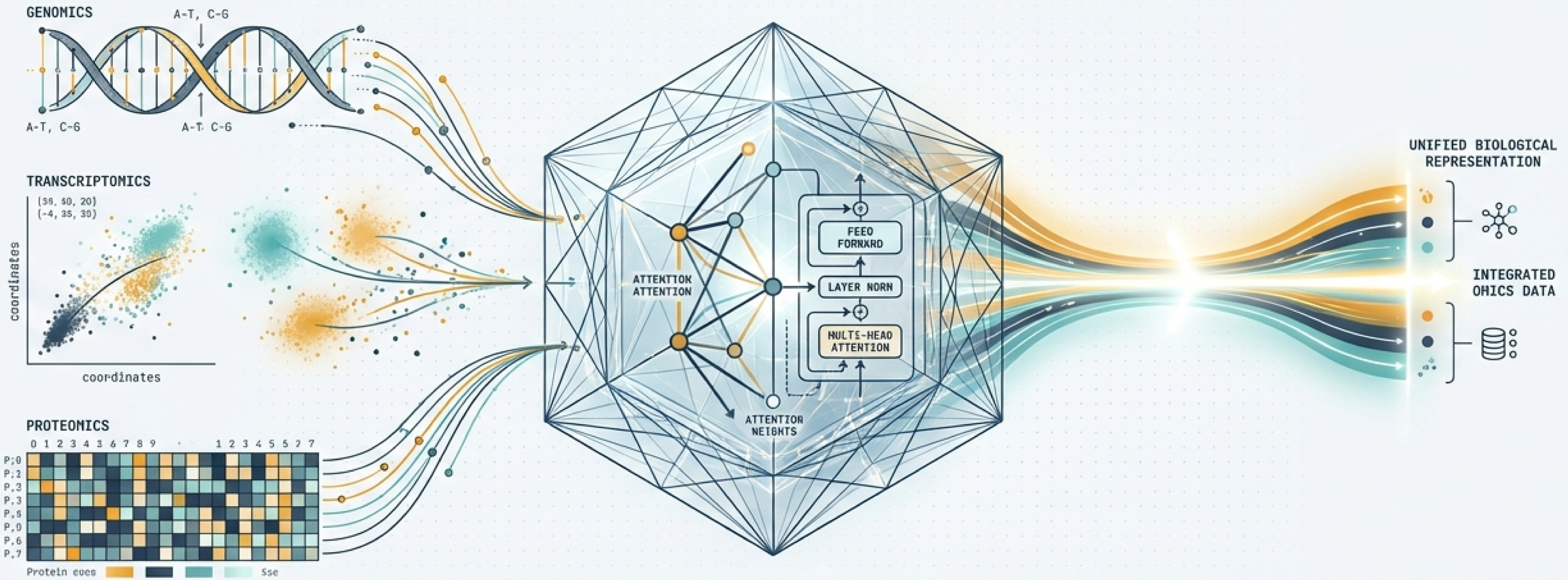
OmicsFormer: Unifying Biological Data with State-of-the-Art Attention Mechanisms



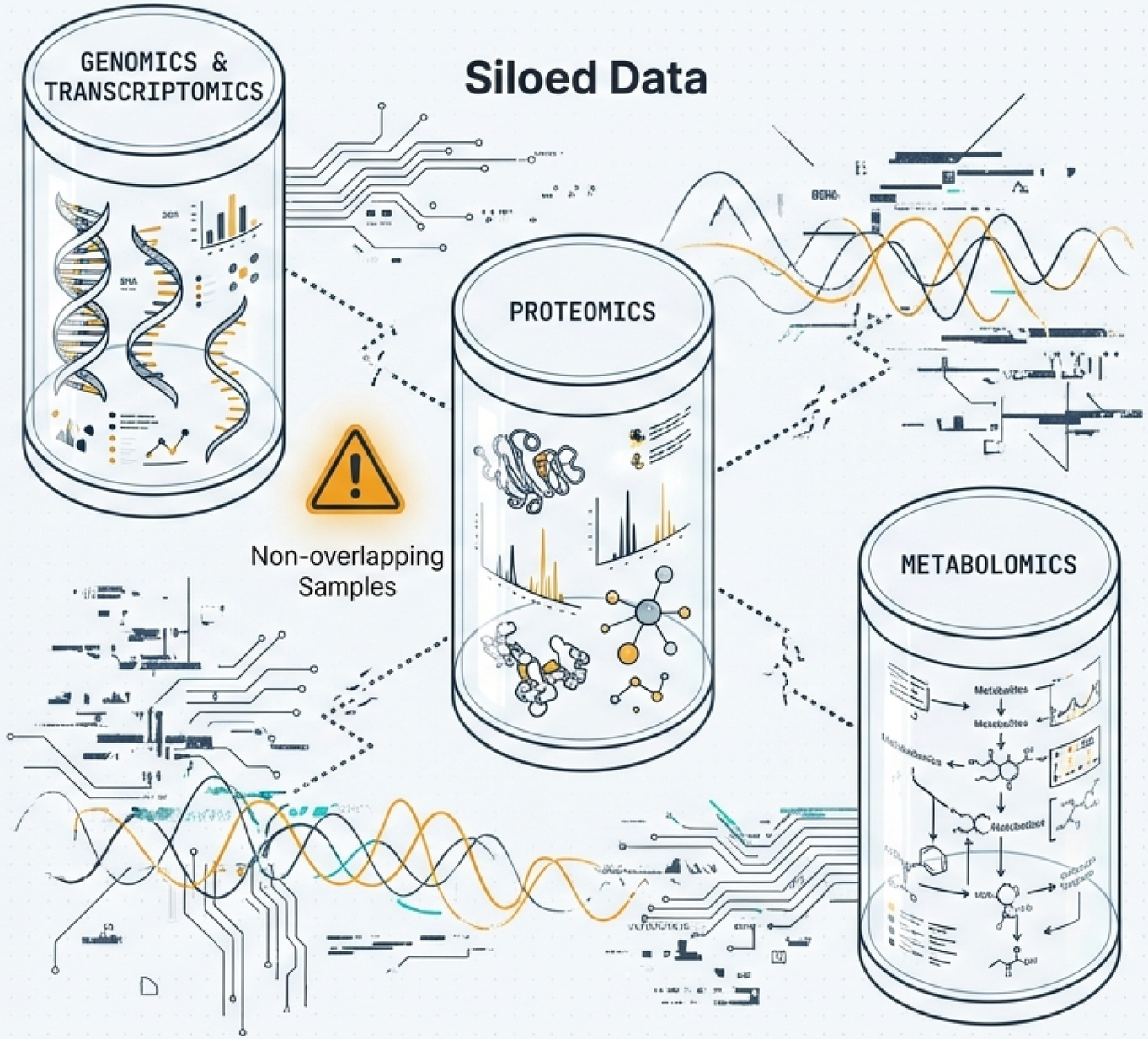
A deep learning framework for integrating multi-omics data, handling missing modalities, and correcting batch effects

2025 | Apache 2.0 License

The Multi-Omics Integration Gap

Biological analysis faces three persistent friction points:

- **Fragmented Modalities:** Genomics, transcriptomics, proteomics, and metabolomics often reside in data silos.
- **Missing Data:** Real-world datasets rarely have perfect overlap, forcing researchers to discard samples.
- **Batch Effects:** Combining independent studies introduces technical noise that obscures biological signals.



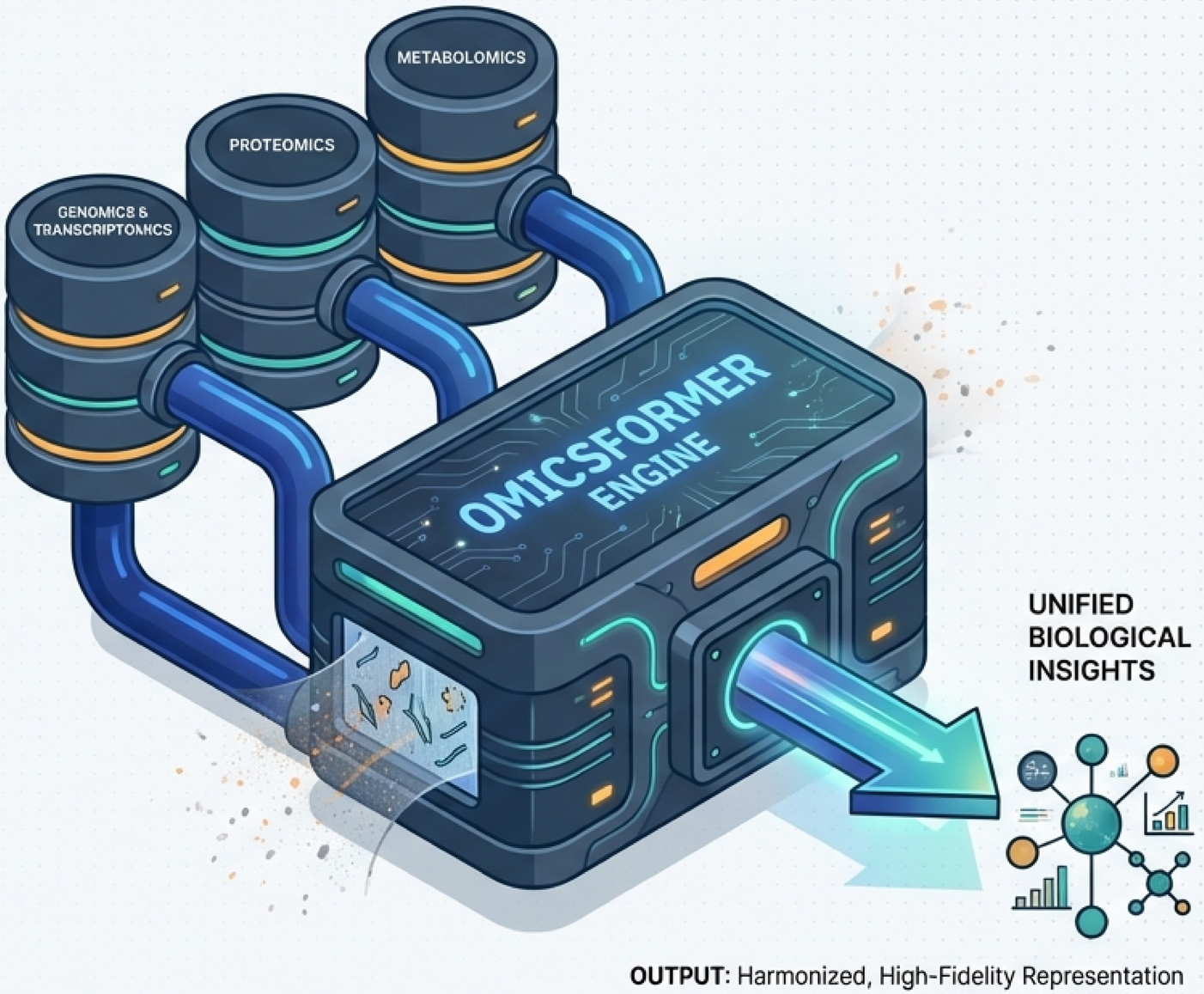
A Unified Deep Learning Framework

OmicsFormer uses transformer architectures to synthesize complex biological data.

Multi-Modal Integration: Seamlessly combines genomics, transcriptomics, proteomics, and metabolomics.

Flexible Alignment: Specifically engineered to handle missing modalities without data loss.

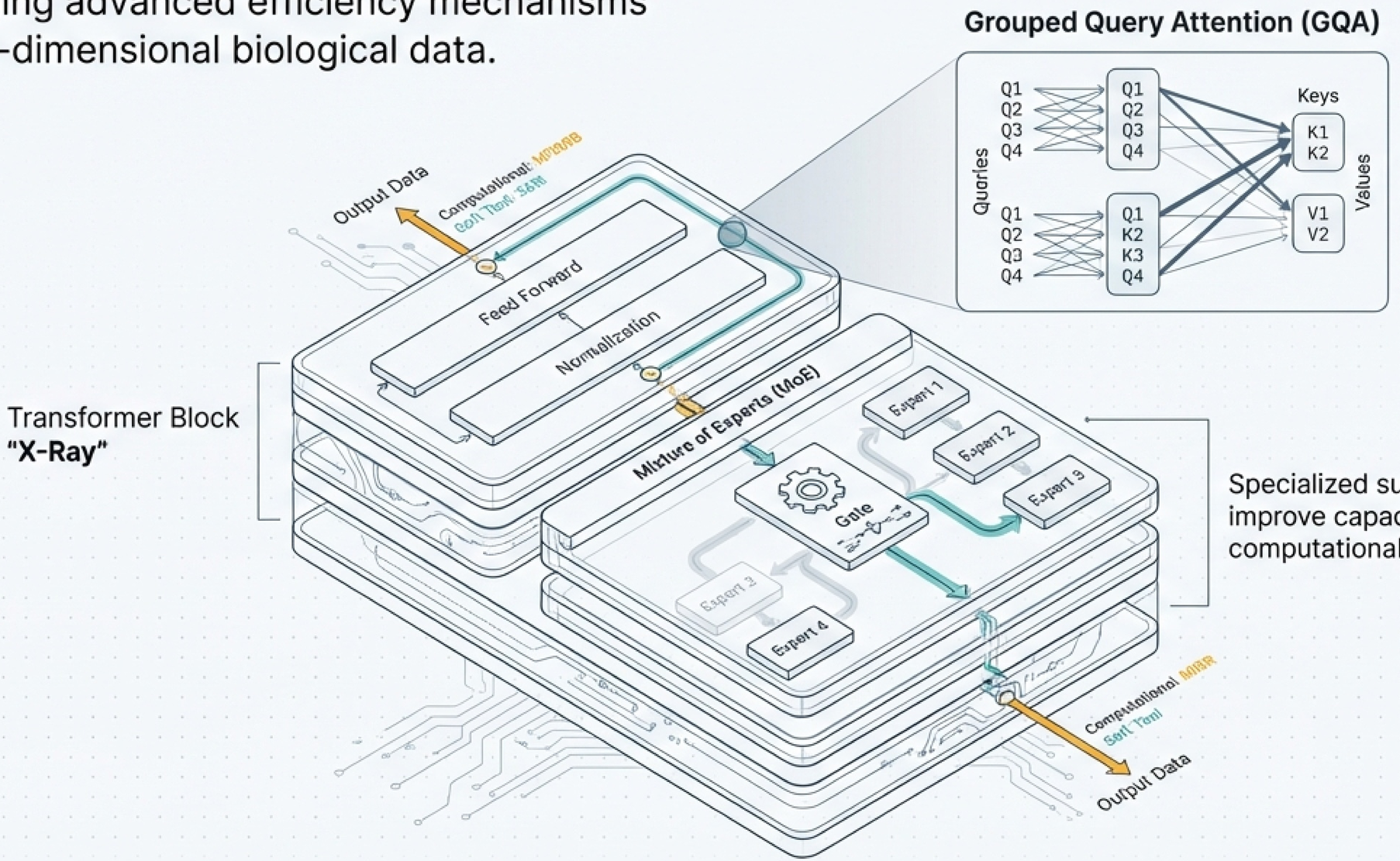
Batch Correction: Natively integrates studies from different platforms and technologies.



OUTPUT: Harmonized, High-Fidelity Representation

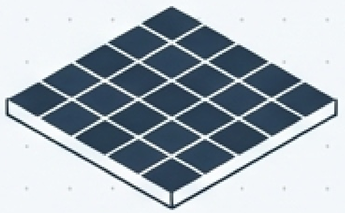
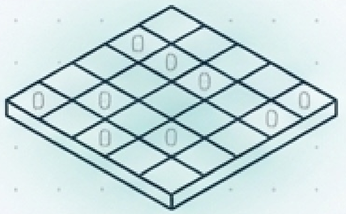
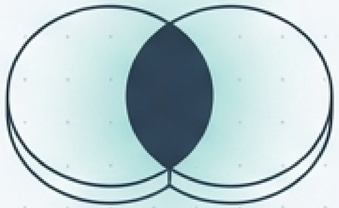
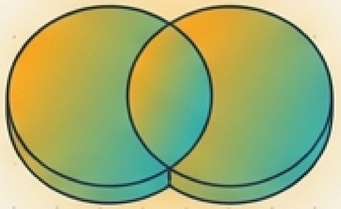
Architecture Built for Efficiency

Leveraging advanced efficiency mechanisms
for high-dimensional biological data.



Four Strategies to Bridge Data Gaps

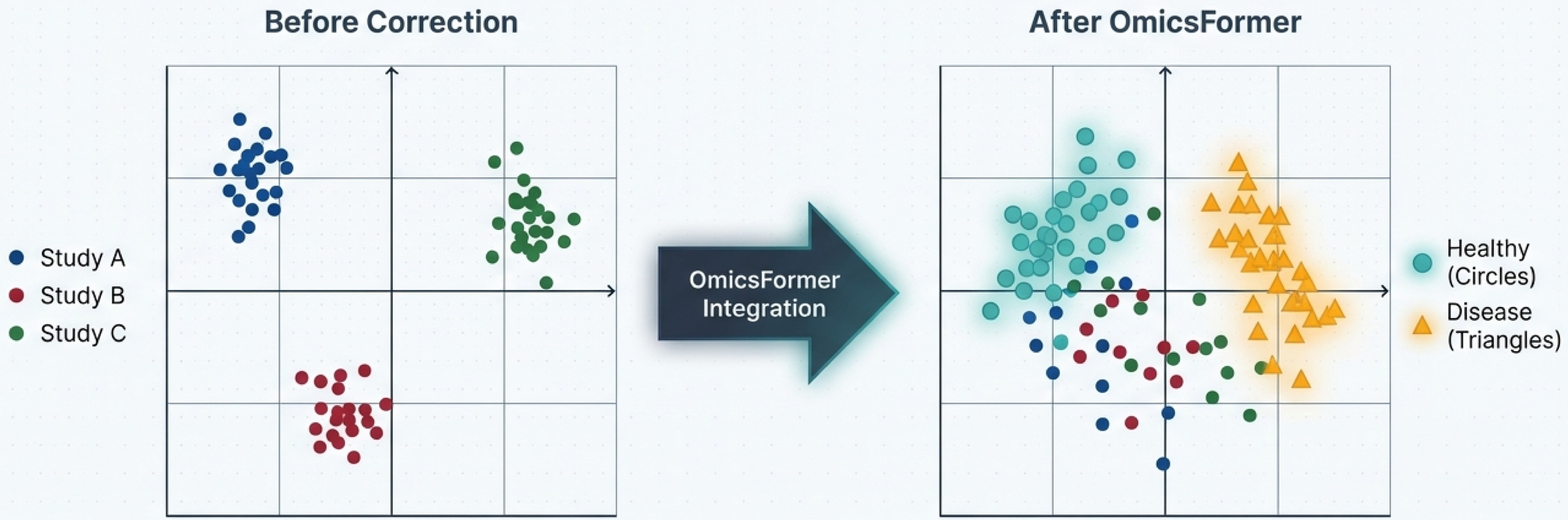
Set 'alignment='auto'' for automatic selection based on data sparsity.

Strict  All modalities required. Validation only.	**Flexible**  Research focus. Missing data is zero-filled.
Intersection  Balanced coverage. Partial overlap allowed.	**Union**  Maximum data utilization. All available data handled.

Eliminating the Batch Effect Noise

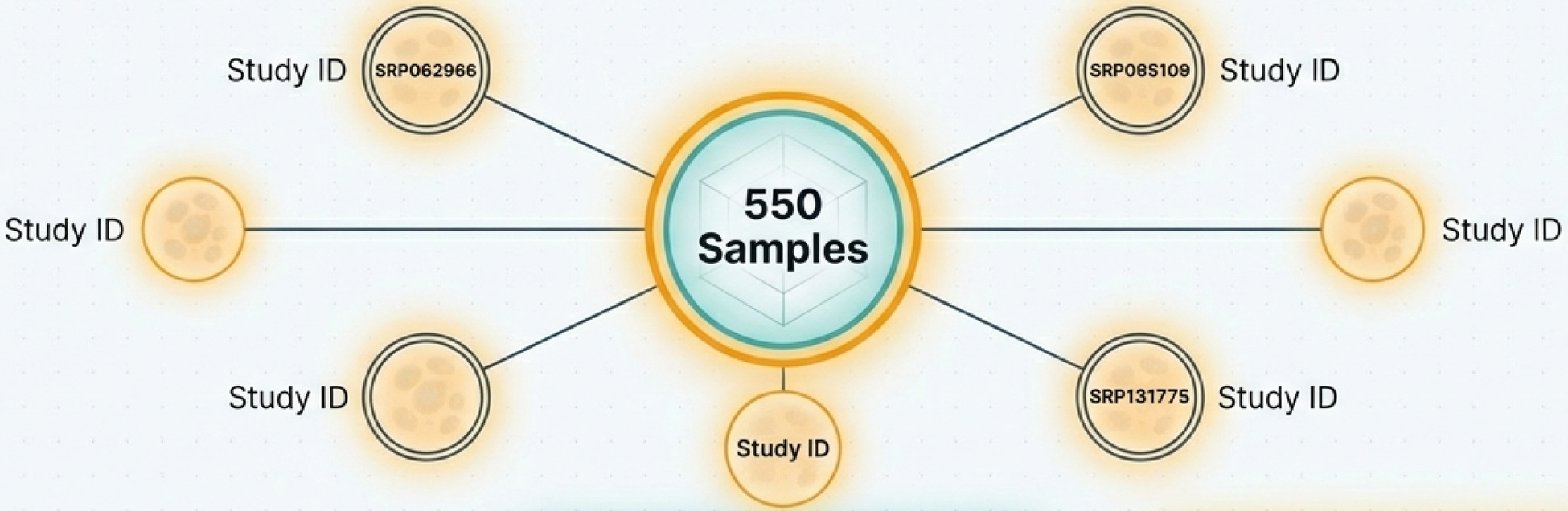
PC1 Variance Reduced: 64.9% → 24.8%

True biological signals emerge when technical noise is removed.



Case Study: Systemic Lupus Erythematosus (SLE)

Large-scale integration of independent datasets.



1. Load 8 Studies

Raw data ingestion

2. Batch Correction (PyCombat)

Batch-effect removal

3. Train Transformer (2008 Inputs)

N=2008 features, Multi-head attention

Verified Performance

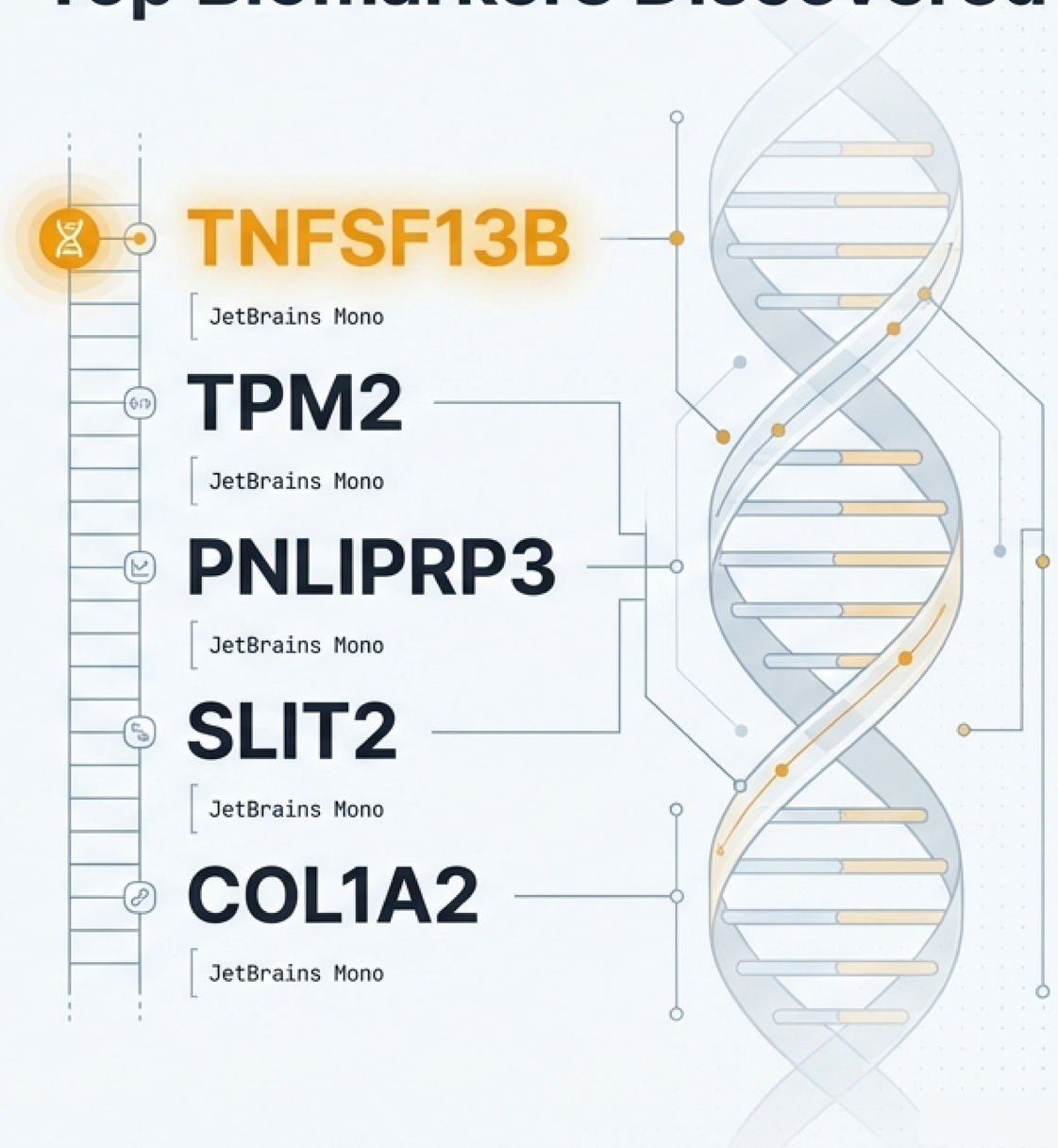
90.91%

Test Accuracy

0.91

F1 Score

Top Biomarkers Discovered



Interpretability: Opening the Black Box

Gradient-based

Measures backpropagation sensitivity.

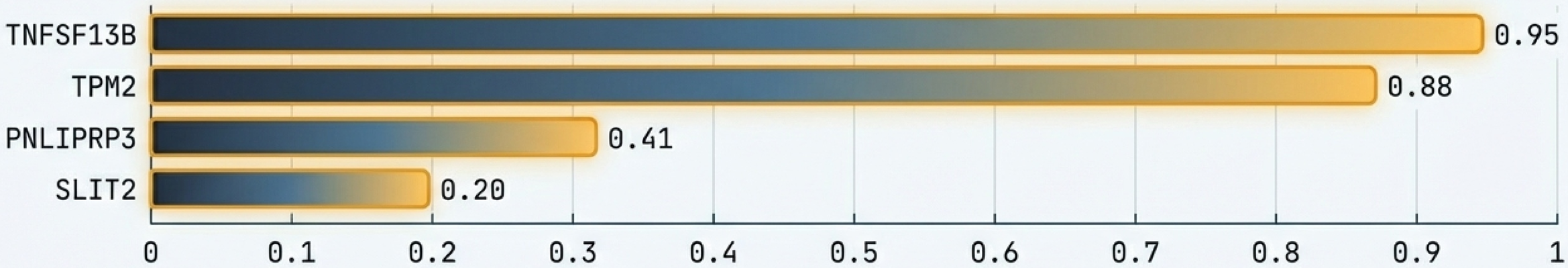
Attention-based

Visualizes model focus patterns.

Permutation-based

Measures impact of scrambling features.

Feature Importance (Scaled 0-1)



All importance scores automatically scaled to [0, 1].

Simple Implementation for Complex Workflows

```python
# 1. Create Dataset
dataset = FlexibleMultiOmicsDataset(
    modality_data={'genomics': genomics_df, 'transcriptomics': rna_df},
    alignment='flexible' # Handles missing modalities
)

# 2. Build Model
model = EnhancedMultiOmicsTransformer(
    input_dims=dataset.feature_dims,
    num_classes=2
)

# 3. Train
trainer = MultiOmicsTrainer(model, train_loader, val_loader)
history = trainer.fit(num_epochs=20)
```

Solves data gaps

Injects SOTA architecture

Applications Across the Clinical Spectrum

Oncology

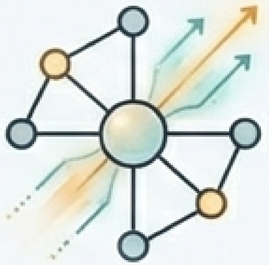
Cancer subtype classification.

Immunology

Biomarker discovery (SLE, RA, IBD).

Pharma

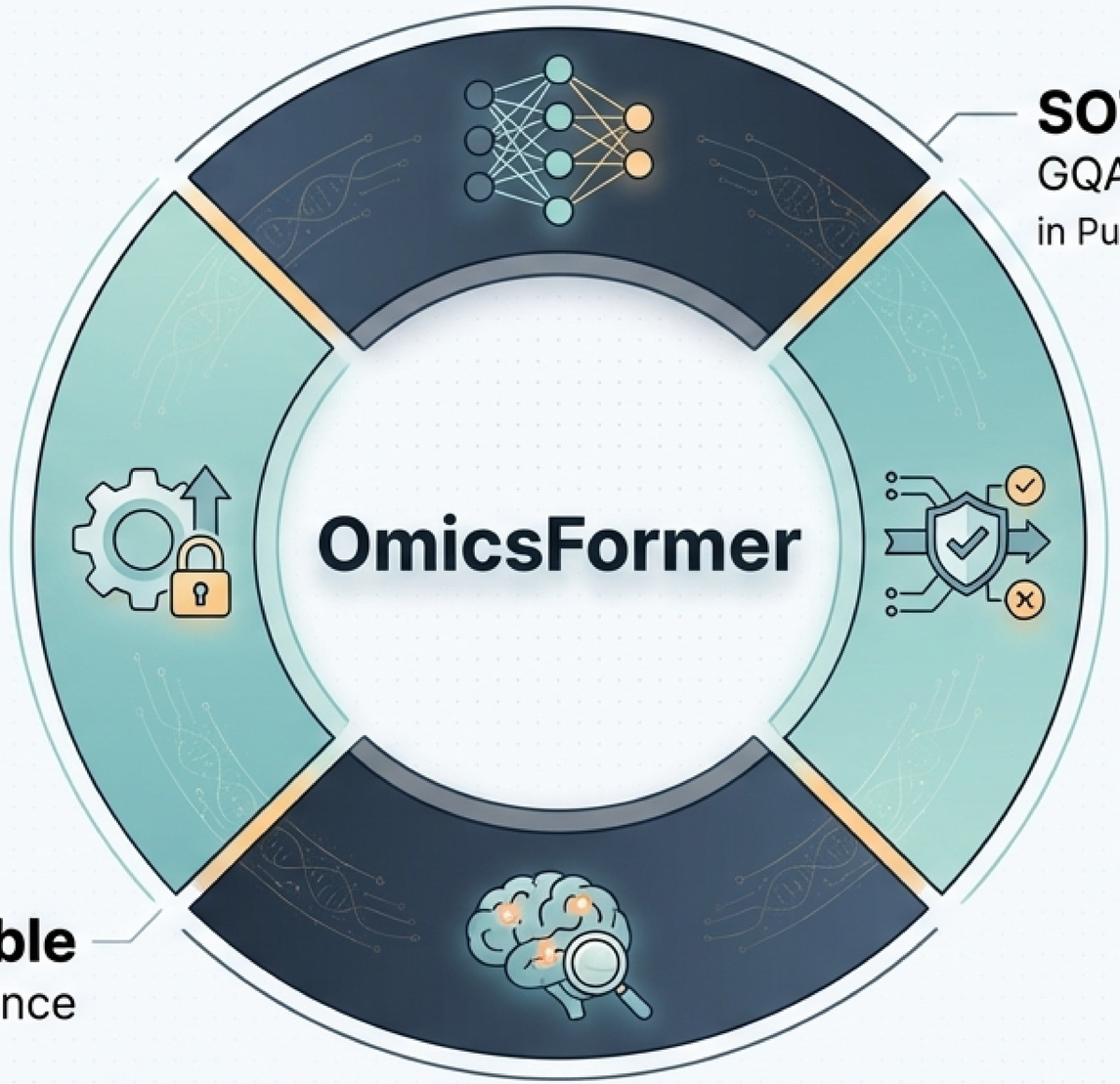
Drug response prediction.



Operations

Cross-platform data harmonization.

The Comprehensive Bioinformatics Solution



SOTA Architecture
GQA & MoE
in Public Sans and JetBrains Mono

Robust
Batch Correction

Interpretable
Feature Importance

Open Source
Apache 2.0

Join the Community

Get the Code

```
pip install -e .
```

GitHub

github.com/shivaprasad-patil/omicsformer

★ 10 3

Patil, S. (2025). OmicsFormer: Multi-Omics Integration with Transformers.

shivaprasad309319@gmail.com