

Python Templates for Neural Image Classification and Spectral Audio Processing

Lightning Hydra Template Extended

Julius O. Smith III
CCRMA, Stanford University

Audio Developer Conference Online (ADCxGather-2025)



github.com/josmithiii/lightning-hydra-template-extended





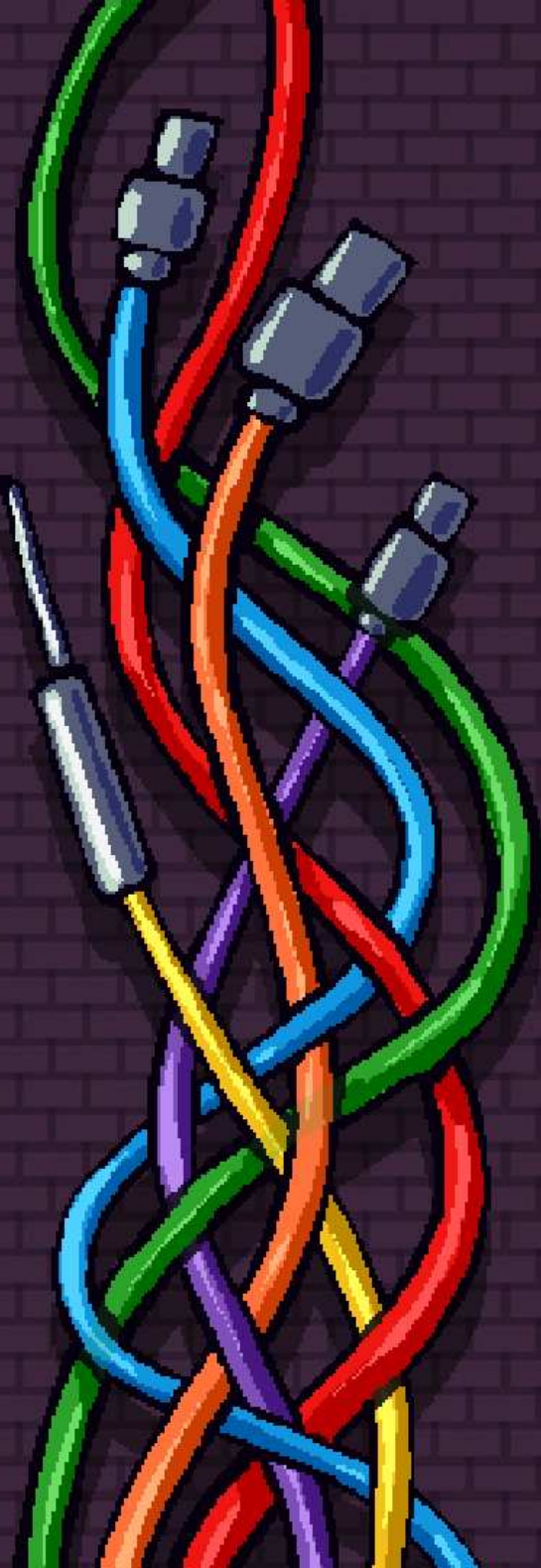
[Lightning Hydra Template *Ext*](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

PyTorch Lightning Hydra Template *Extended* (LHTE)



ADCx GATHER

PYTHON TEMPLATES FOR NEURAL IMAGE
CLASSIFICATION AND SPECTRAL AUDIO
PROCESSING

LIGHTNING HYDRA TEMPLATE EXTENDED

JULIUS SMITH



[Lightning Hydra Template Ext](#)

- [Lightning Hydra Template Ext](#)
- [Original LHT](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**



Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- Original LHT

New Dataset Formats

New Architectures

Workflows

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**
- LHTE began as a fork of the *Lightning Hydra Template (LHT)*:
<https://github.com/ashleve/lightning-hydra-template>



Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- Original LHT

New Dataset Formats

New Architectures

Workflows

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**
- LHTE began as a fork of the *Lightning Hydra Template (LHT)*:
<https://github.com/ashleve/lightning-hydra-template>
 - *PyTorch Lightning* streamlines deep-learning model development in numerous ways



Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- Original LHT

New Dataset Formats

New Architectures

Workflows

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**
- LHTE began as a fork of the *Lightning Hydra Template (LHT)*:
<https://github.com/ashleve/lightning-hydra-template>
 - *PyTorch Lightning* streamlines deep-learning model development in numerous ways
 - *Hydra* is a powerful *configuration management framework* often used



Lightning Hydra Template Ext

- [Lightning Hydra Template Ext](#)
- [Original LHT](#)

New Dataset Formats

New Architectures

Workflows

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**
- LHTE began as a fork of the *Lightning Hydra Template (LHT)*:
<https://github.com/ashleve/lightning-hydra-template>
 - *PyTorch Lightning* streamlines deep-learning model development in numerous ways
 - *Hydra* is a powerful *configuration management framework* often used
 - An LHT *example experiment* trains an MLP on MNIST (hand-written digits)



Lightning Hydra Template Ext

- [Lightning Hydra Template Ext](#)
- [Original LHT](#)

New Dataset Formats

New Architectures

Workflows

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**
- LHTE began as a fork of the *Lightning Hydra Template (LHT)*:
<https://github.com/ashleve/lightning-hydra-template>
 - *PyTorch Lightning* streamlines deep-learning model development in numerous ways
 - *Hydra* is a powerful *configuration management framework* often used
 - An LHT *example experiment* trains an MLP on MNIST (hand-written digits)
- LHTE adds *more architectures*: CNNs, ViT, ConvNeXt, EfficientNet



Lightning Hydra Template Ext

- [Lightning Hydra Template Ext](#)
- [Original LHT](#)

New Dataset Formats

New Architectures

Workflows

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**
- LHTE began as a fork of the *Lightning Hydra Template (LHT)*:
<https://github.com/ashleve/lightning-hydra-template>
 - *PyTorch Lightning* streamlines deep-learning model development in numerous ways
 - *Hydra* is a powerful *configuration management framework* often used
 - An LHT *example experiment* trains an MLP on MNIST (hand-written digits)
- LHTE adds *more architectures*: CNNs, ViT, ConvNeXt, EfficientNet
- LHTE adds *more datasets*: CIFAR-10, CIFAR-100, VIMH:



Lightning Hydra Template Ext

- [Lightning Hydra Template Ext](#)
- [Original LHT](#)

New Dataset Formats

New Architectures

Workflows

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**
- LHTE began as a fork of the *Lightning Hydra Template (LHT)*:
<https://github.com/ashleve/lightning-hydra-template>
 - *PyTorch Lightning* streamlines deep-learning model development in numerous ways
 - *Hydra* is a powerful *configuration management framework* often used
 - An LHT *example experiment* trains an MLP on MNIST (hand-written digits)
- LHTE adds *more architectures*: CNNs, ViT, ConvNeXt, EfficientNet
- LHTE adds *more datasets*: CIFAR-10, CIFAR-100, VIMH:
- Variable Image Multi-Head (VIMH) format generalizes CIFAR-100:



Lightning Hydra Template Ext

- [Lightning Hydra Template Ext](#)
- [Original LHT](#)

New Dataset Formats

New Architectures

Workflows

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**
- LHTE began as a fork of the *Lightning Hydra Template (LHT)*:
<https://github.com/ashleve/lightning-hydra-template>
 - *PyTorch Lightning* streamlines deep-learning model development in numerous ways
 - *Hydra* is a powerful *configuration management framework* often used
 - An LHT *example experiment* trains an MLP on MNIST (hand-written digits)
- LHTE adds *more architectures*: CNNs, ViT, ConvNeXt, EfficientNet
- LHTE adds *more datasets*: CIFAR-10, CIFAR-100, VIMH:
- Variable Image Multi-Head (VIMH) format generalizes CIFAR-100:
 - Up to 65k x 65k images with up to 65k channels



Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- Original LHT

New Dataset Formats

New Architectures

Workflows

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**
- LHTE began as a fork of the *Lightning Hydra Template (LHT)*:
<https://github.com/ashleve/lightning-hydra-template>
 - *PyTorch Lightning* streamlines deep-learning model development in numerous ways
 - *Hydra* is a powerful *configuration management framework* often used
 - An LHT *example experiment* trains an MLP on MNIST (hand-written digits)
- LHTE adds *more architectures*: CNNs, ViT, ConvNeXt, EfficientNet
- LHTE adds *more datasets*: CIFAR-10, CIFAR-100, VIMH:
- Variable Image Multi-Head (VIMH) format generalizes CIFAR-100:
 - Up to 65k x 65k images with up to 65k channels
 - *Multi-Head ready*: Up to 65k *labeled parameters* (8-bit label, 8-bit value)



Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- Original LHT

New Dataset Formats

New Architectures

Workflows

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**
- LHTE began as a fork of the *Lightning Hydra Template (LHT)*:
<https://github.com/ashleve/lightning-hydra-template>
 - *PyTorch Lightning* streamlines deep-learning model development in numerous ways
 - *Hydra* is a powerful *configuration management framework* often used
 - An LHT *example experiment* trains an MLP on MNIST (hand-written digits)
- LHTE adds *more architectures*: CNNs, ViT, ConvNeXt, EfficientNet
- LHTE adds *more datasets*: CIFAR-10, CIFAR-100, VIMH:
- Variable Image Multi-Head (VIMH) format generalizes CIFAR-100:
 - Up to 65k x 65k images with up to 65k channels
 - *Multi-Head ready*: Up to 65k *labeled parameters* (8-bit label, 8-bit value)
 - Example Use: *synthesis parameters* (≤ 256 parameters with ≤ 256 settings each)





Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- Original LHT

New Dataset Formats

New Architectures

Workflows

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**
- LHTE began as a fork of the *Lightning Hydra Template (LHT)*:
<https://github.com/ashleve/lightning-hydra-template>
 - *PyTorch Lightning* streamlines deep-learning model development in numerous ways
 - *Hydra* is a powerful *configuration management framework* often used
 - An LHT *example experiment* trains an MLP on MNIST (hand-written digits)
- LHTE adds *more architectures*: CNNs, ViT, ConvNeXt, EfficientNet
- LHTE adds *more datasets*: CIFAR-10, CIFAR-100, VIMH:
- Variable Image Multi-Head (VIMH) format generalizes CIFAR-100:
 - Up to 65k x 65k images with up to 65k channels
 - *Multi-Head ready*: Up to 65k *labeled parameters* (8-bit label, 8-bit value)
 - Example Use: *synthesis parameters* (≤ 256 parameters with ≤ 256 settings each)
 - *Data loaders* for all supported datasets and model architectures





Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- Original LHT

New Dataset Formats

New Architectures

Workflows

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**
- LHTE began as a fork of the *Lightning Hydra Template (LHT)*:
<https://github.com/ashleve/lightning-hydra-template>
 - *PyTorch Lightning* streamlines deep-learning model development in numerous ways
 - *Hydra* is a powerful *configuration management framework* often used
 - An LHT *example experiment* trains an MLP on MNIST (hand-written digits)
- LHTE adds *more architectures*: CNNs, ViT, ConvNeXt, EfficientNet
- LHTE adds *more datasets*: CIFAR-10, CIFAR-100, VIMH:
- Variable Image Multi-Head (VIMH) format generalizes CIFAR-100:
 - Up to 65k x 65k images with up to 65k channels
 - *Multi-Head ready*: Up to 65k *labeled parameters* (8-bit label, 8-bit value)
 - Example Use: *synthesis parameters* (≤ 256 parameters with ≤ 256 settings each)
 - *Data loaders* for all supported datasets and model architectures
 - The `configs/experiment/` directory contains *benchmark replications*





Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- Original LHT

New Dataset Formats

New Architectures

Workflows

Lightning Hydra Template Extended (LHTE) — New Architectures and Datasets

- Extends the **Pytorch Lightning + Hydra Template** with top **image-classification architectures** and **additional dataset formats**
- LHTE began as a fork of the *Lightning Hydra Template (LHT)*:
<https://github.com/ashleve/lightning-hydra-template>
 - *PyTorch Lightning* streamlines deep-learning model development in numerous ways
 - *Hydra* is a powerful *configuration management framework* often used
 - An LHT *example experiment* trains an MLP on MNIST (hand-written digits)
- LHTE adds *more architectures*: CNNs, ViT, ConvNeXt, EfficientNet
- LHTE adds *more datasets*: CIFAR-10, CIFAR-100, VIMH:
- Variable Image Multi-Head (VIMH) format generalizes CIFAR-100:
 - Up to 65k x 65k images with up to 65k channels
 - *Multi-Head ready*: Up to 65k *labeled parameters* (8-bit label, 8-bit value)
 - Example Use: *synthesis parameters* (≤ 256 parameters with ≤ 256 settings each)
 - *Data loaders* for all supported datasets and model architectures
 - The `configs/experiment/` directory contains *benchmark replications*
 - Top-level Makefile contains ≈ 80 *make targets* for tests, training, experiments





Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- [Original LHT](#)

New Dataset Formats

New Architectures

Workflows

Original Lightning Hydra Template (LHT)

Open-Source PyTorch Lightning + Hydra Bootstrap

<https://github.com/ashleve/lightning-hydra-template>



Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- [Original LHT](#)

New Dataset Formats

New Architectures

Workflows

Original Lightning Hydra Template (LHT)

Open-Source PyTorch Lightning + Hydra Bootstrap

<https://github.com/ashleve/lightning-hydra-template>

- PyTorch *project skeleton* with `src/`, `configs/`, and `tests/`, ready for research



Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- [Original LHT](#)

New Dataset Formats

New Architectures

Workflows

Original Lightning Hydra Template (LHT)

Open-Source PyTorch Lightning + Hydra Bootstrap

<https://github.com/ashleve/lightning-hydra-template>

- PyTorch *project skeleton* with `src/`, `configs/`, and `tests/`, ready for research
- *Chatbots love it*



Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- **Original LHT**

New Dataset Formats

New Architectures

Workflows

Original Lightning Hydra Template (LHT)

Open-Source PyTorch Lightning + Hydra Bootstrap

<https://github.com/ashleve/lightning-hydra-template>

- PyTorch *project skeleton* with `src/`, `configs/`, and `tests/`, ready for research
- *Chatbots love it*
- *Hydra workflow*: CLI overrides such as:
 `> python src/train.py trainer=gpu data.batch_size=64`
for every configuration value (≈ 300 — thousands in the LHTE)



Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- [Original LHT](#)

New Dataset Formats

New Architectures

Workflows

Original Lightning Hydra Template (LHT)

Open-Source PyTorch Lightning + Hydra Bootstrap

<https://github.com/ashleve/lightning-hydra-template>

- PyTorch *project skeleton* with `src/`, `configs/`, and `tests/`, ready for research
- *Chatbots love it*
- *Hydra workflow*: CLI overrides such as:
 `> python src/train.py trainer=gpu data.batch_size=64`
 for every configuration value (≈ 300 — thousands in the LHTE)
- *PyTorch Lightning baseline*: single-head MNIST MLP experiment — *that's it!*



Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- **Original LHT**

New Dataset Formats

New Architectures

Workflows

Original Lightning Hydra Template (LHT)

Open-Source PyTorch Lightning + Hydra Bootstrap

<https://github.com/ashleve/lightning-hydra-template>

- PyTorch *project skeleton* with `src/`, `configs/`, and `tests/`, ready for research
- *Chatbots love it*
- *Hydra workflow*: CLI overrides such as:
 `> python src/train.py trainer=gpu data.batch_size=64`
 for every configuration value (≈ 300 — thousands in the LHTE)
- *PyTorch Lightning baseline*: single-head MNIST MLP experiment — *that's it!*
- *Make targets* for testing, training, and admin



Lightning Hydra Template Ext

- Lightning Hydra Template Ext
- [Original LHT](#)

New Dataset Formats

New Architectures

Workflows

Original Lightning Hydra Template (LHT)

Open-Source PyTorch Lightning + Hydra Bootstrap

<https://github.com/ashleve/lightning-hydra-template>

- PyTorch *project skeleton* with `src/`, `configs/`, and `tests/`, ready for research
- *Chatbots love it*
- *Hydra workflow*: CLI overrides such as:
 `> python src/train.py trainer=gpu data.batch_size=64`
 for every configuration value (≈ 300 — thousands in the LHTE)
- *PyTorch Lightning baseline*: single-head MNIST MLP experiment — *that's it!*
- *Make targets* for testing, training, and admin
- LHTE maintains *compatibility* with existing LHT configs, datasets, and checkpoints



[Lightning Hydra Template *Ext*](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

Datasets Beyond MNIST



[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

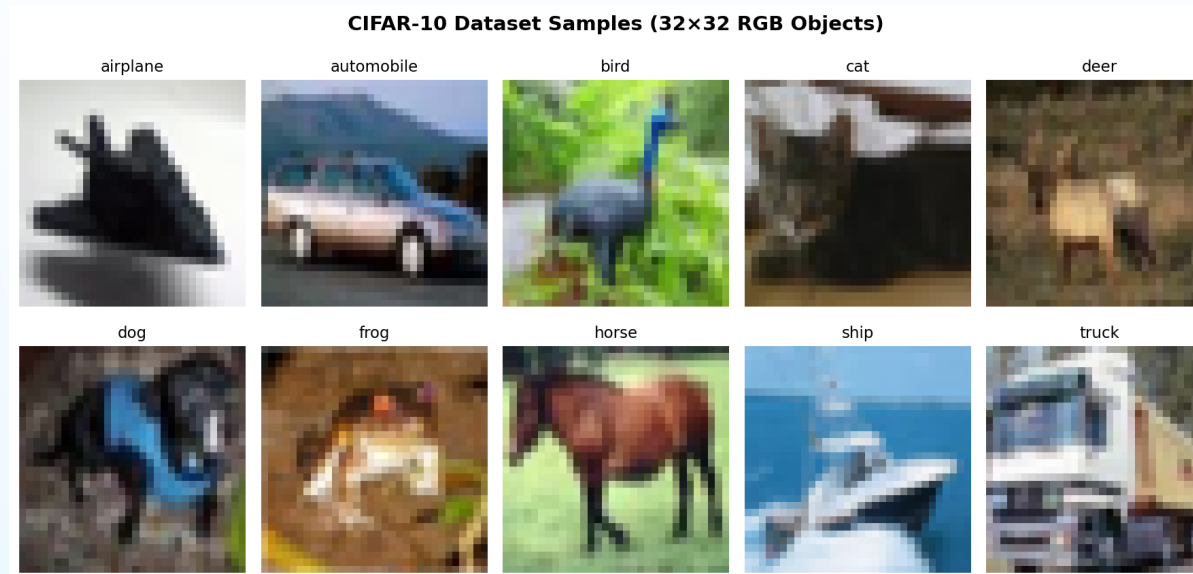
- [CIFAR-10](#)
- CIFAR-100
- VIMH

[New Architectures](#)

[Workflows](#)

CIFAR-10 Dataset Support

Standard Computer Vision Benchmark



- 10 classes, 32×32×3 (RGB), 50K train / 10K test
- *LHTE Results:* CNN (85-92%), ViT (88-93%), ConvNeXt (90-95%)
- *CIFAR-10 CNN Benchmark Example (also have MLP, ViT, ConvNeXt, EfficientNet, ...):*
 - make cbq10c (CIFAR-10 Benchmark Quick, CNN)
 - make cb10c (CIFAR-10 Benchmark, CNN)
- Literature-reproducing performance out-of-the-box



[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

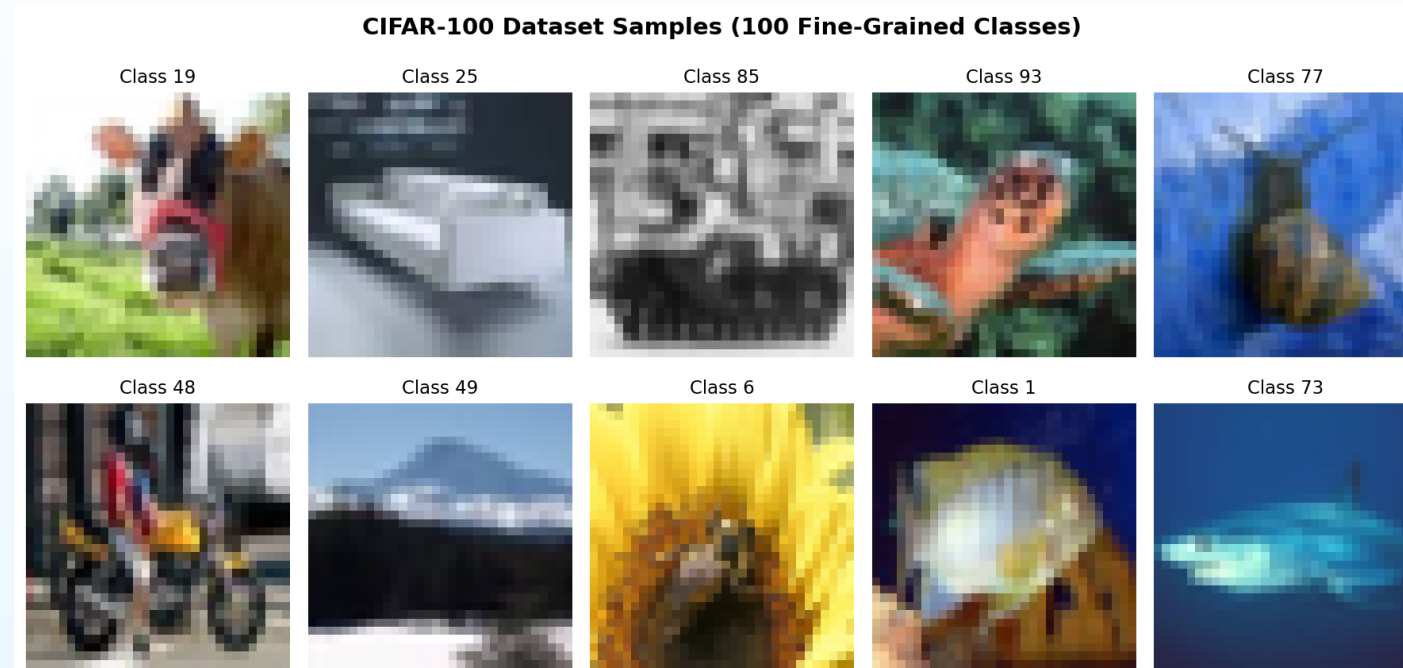
- [CIFAR-10](#)
- [CIFAR-100](#)
- [VIMH](#)

[New Architectures](#)

[Workflows](#)

CIFAR-100 Dataset Support

Fine-Grained Classification Challenge



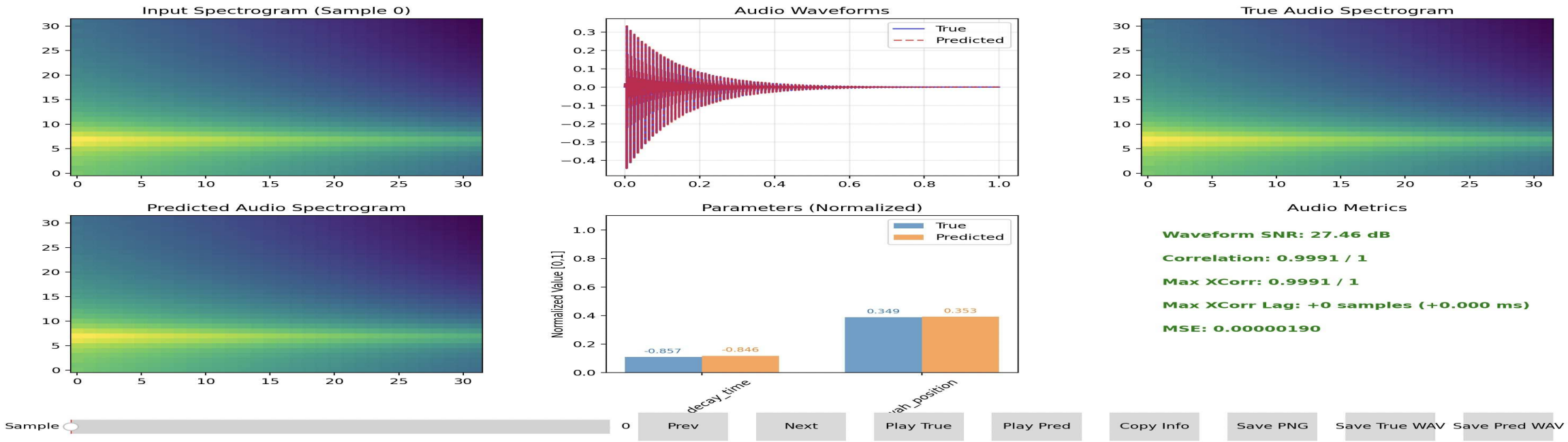
- 100 fine classes or 20 coarse superclasses, 32×32×3 RGB
- *Performance*: ConvNeXt leads at 70-80% (fine), 85% (coarse)
- *Multihead*: Simultaneous fine + coarse classification
- *CIFAR-100 Benchmark Commands* (say `make help` for the complete list):
 - `make cb100c` (or `make cifar100-cnn`: 55-70% expected accuracy)
 - `make cbs100` (run *all* CIFAR-100 benchmarks — the “benchmark suite”))



Variable Image MultiHead (VIMH) Format

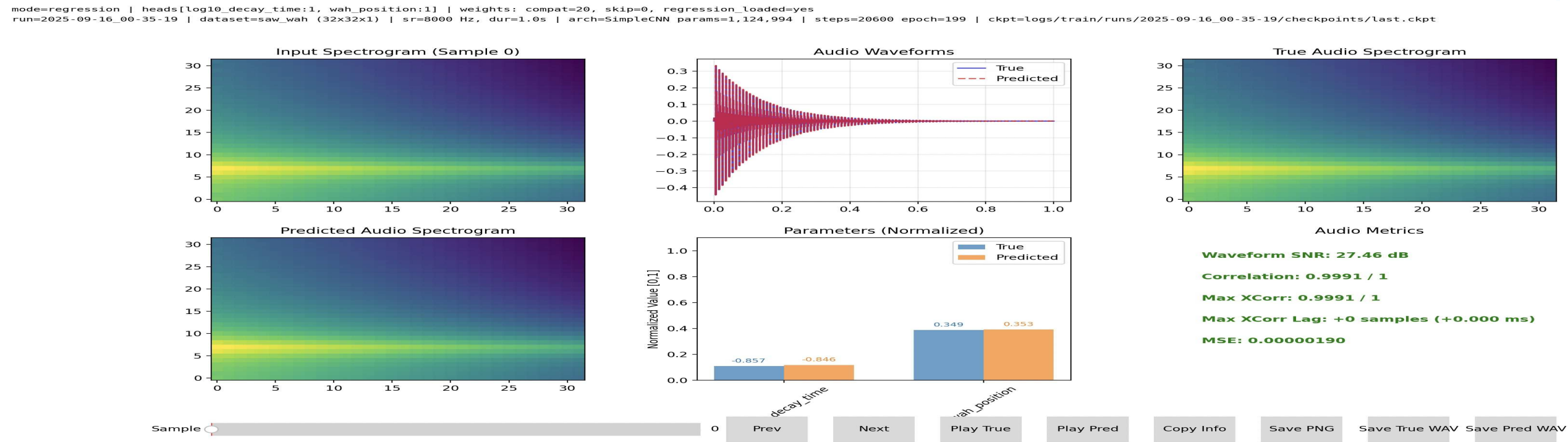
Generalized Dataset Format for Multi-Parameter Learning (Multi-Category Classification or Regression)

mode=regression | heads[log10_decay_time:1, wah_position:1] | weights: compat=20, skip=0, regression_loaded=yes
run=2025-09-16_00-35-19 | dataset=saw_wah (32x32x1) | sr=8000 Hz, dur=1.0s | arch=SimpleCNN params=1,124,994 | steps=20600 epoch=199 | ckpt=logs/train/runs/2025-09-16_00-35-19/checkpoints/last.ckpt



Variable Image MultiHead (VIMH) Format

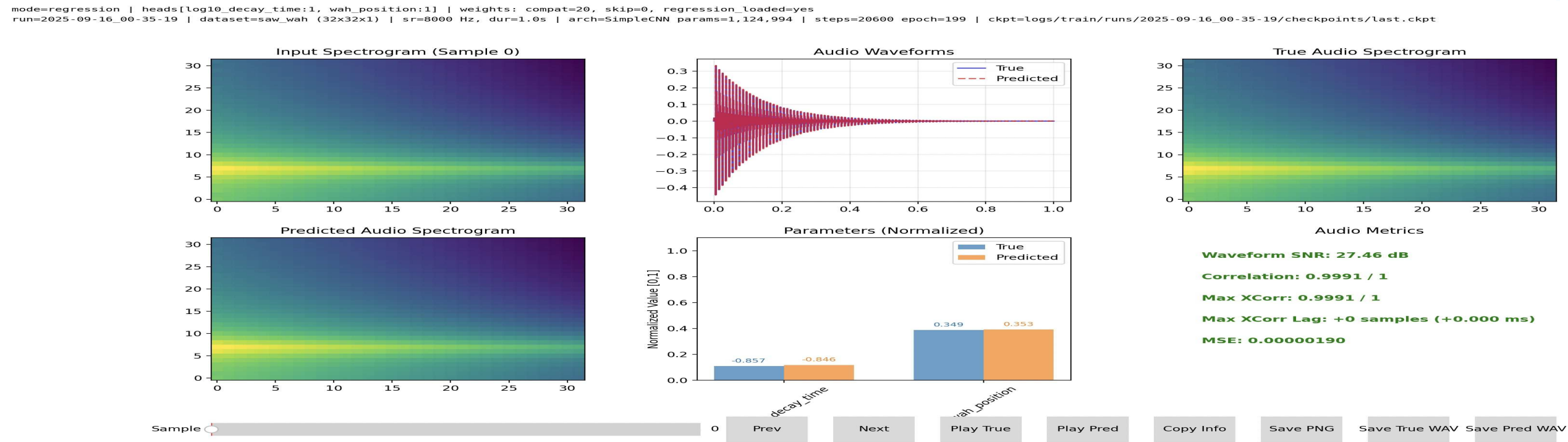
Generalized Dataset Format for Multi-Parameter Learning (Multi-Category Classification or Regression)



- *Self-describing:* Variable dimensions up to 65k×65k×65k (16-bit metadata)

Variable Image MultiHead (VIMH) Format

Generalized Dataset Format for Multi-Parameter Learning (Multi-Category Classification or Regression)

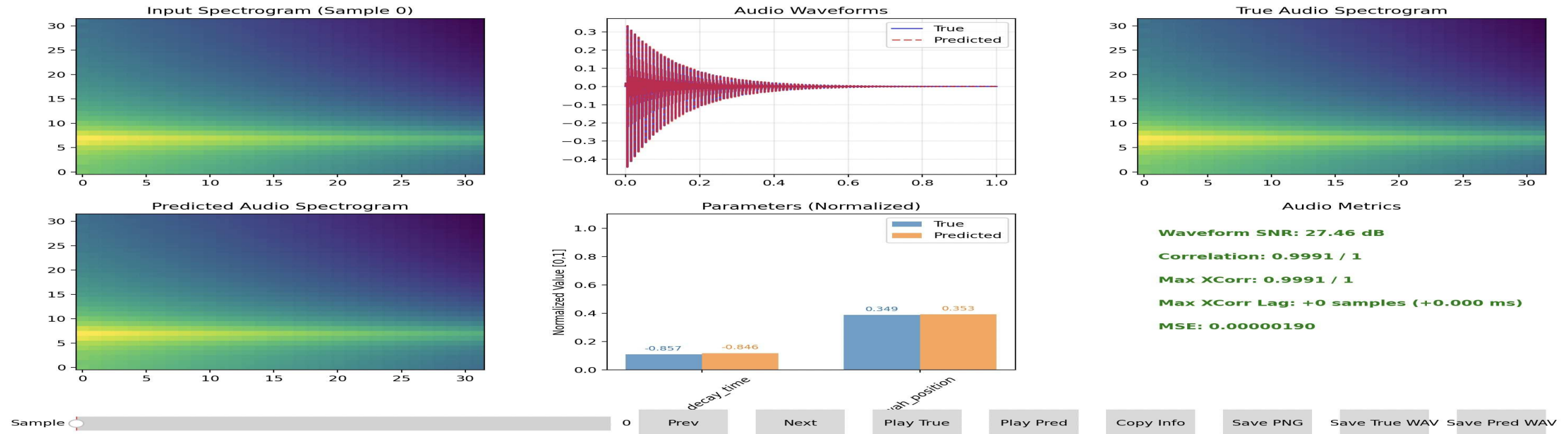


- *Self-describing:* Variable dimensions up to 65k×65k×65k (16-bit metadata)
- *Flexible labels:* 0-255 parameters, 8-bit quantized (100 steps)

Variable Image MultiHead (VIMH) Format

Generalized Dataset Format for Multi-Parameter Learning (Multi-Category Classification or Regression)

```
mode=regression | heads[log10_decay_time:1, wah_position:1] | weights: compat=20, skip=0, regression_loaded=yes  
run=2025-09-16_00-35-19 | dataset=saw_wah (32x32x1) | sr=8000 Hz, dur=1.0s | arch=SimpleCNN params=1,124,994 | steps=20600 epoch=199 | ckpt=logs/train/runs/2025-09-16_00-35-19/checkpoints/last.ckpt
```

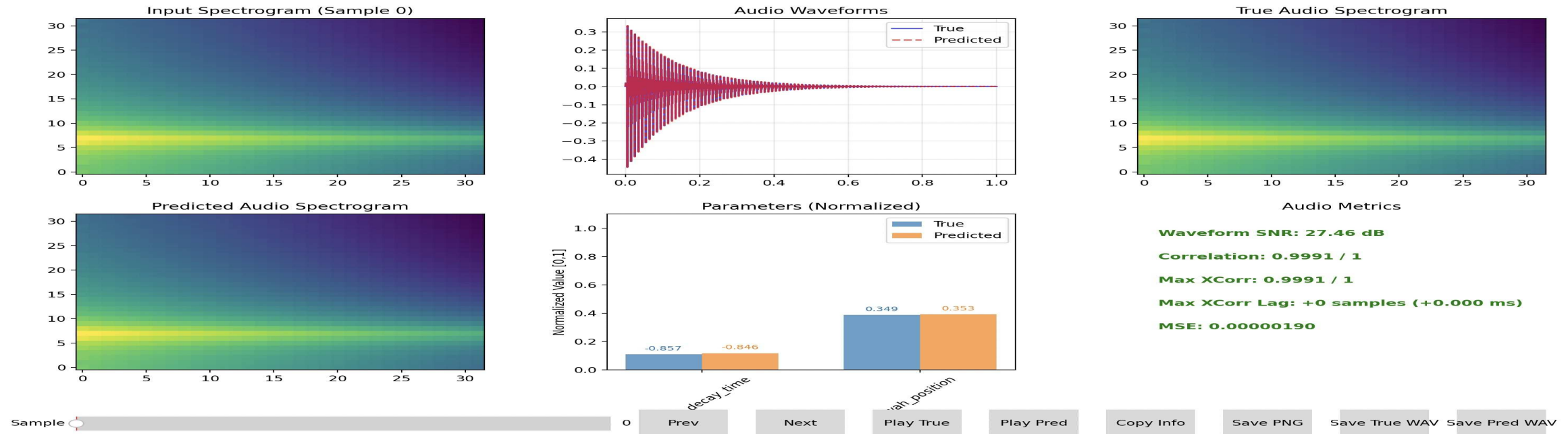


- *Self-describing:* Variable dimensions up to 65k×65k×65k (16-bit metadata)
- *Flexible labels:* 0-255 parameters, 8-bit quantized (100 steps)
- *Per sample:* [height, width, channels] + [N params] + [ID,val pairs] + image

Variable Image MultiHead (VIMH) Format

Generalized Dataset Format for Multi-Parameter Learning (Multi-Category Classification or Regression)

```
mode=regression | heads[log10_decay_time:1, wah_position:1] | weights: compat=20, skip=0, regression_loaded=yes  
run=2025-09-16_00-35-19 | dataset=saw_wah (32x32x1) | sr=8000 Hz, dur=1.0s | arch=SimpleCNN params=1,124,994 | steps=20600 epoch=199 | ckpt=logs/train/runs/2025-09-16_00-35-19/checkpoints/last.ckpt
```

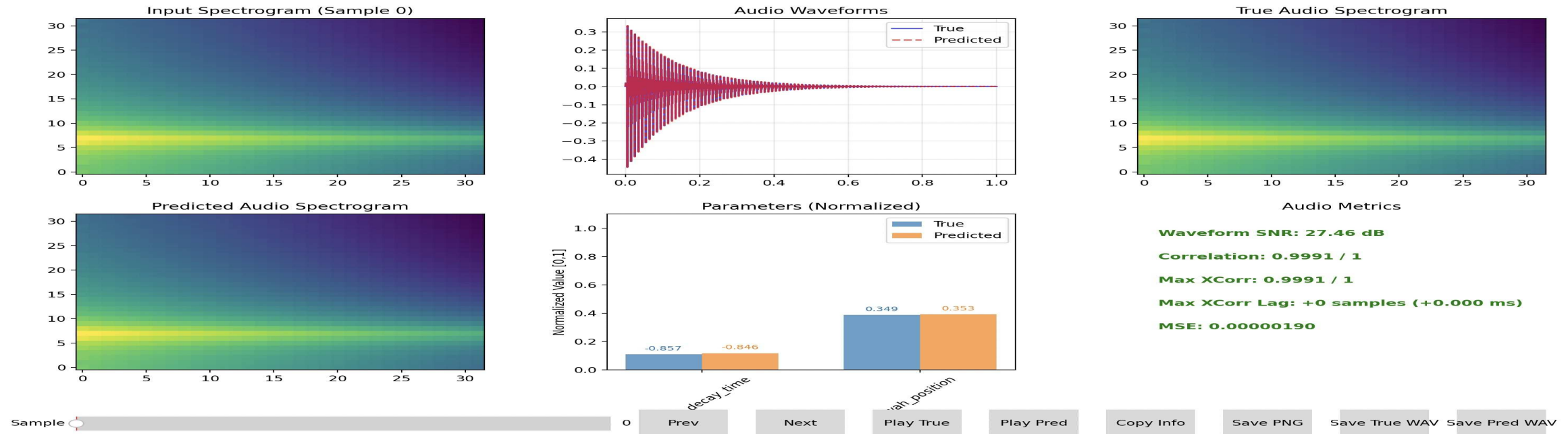


- *Self-describing:* Variable dimensions up to 65k×65k×65k (16-bit metadata)
- *Flexible labels:* 0-255 parameters, 8-bit quantized (100 steps)
- *Per sample:* [height, width, channels] + [N params] + [ID,val pairs] + image
- *Applications:* Audio spectrograms, vision, scientific data

Variable Image MultiHead (VIMH) Format

Generalized Dataset Format for Multi-Parameter Learning (Multi-Category Classification or Regression)

```
mode=regression | heads[log10_decay_time:1, wah_position:1] | weights: compat=20, skip=0, regression_loaded=yes  
run=2025-09-16_00-35-19 | dataset=saw_wah (32x32x1) | sr=8000 Hz, dur=1.0s | arch=SimpleCNN params=1,124,994 | steps=20600 epoch=199 | ckpt=logs/train/runs/2025-09-16_00-35-19/checkpoints/last.ckpt
```

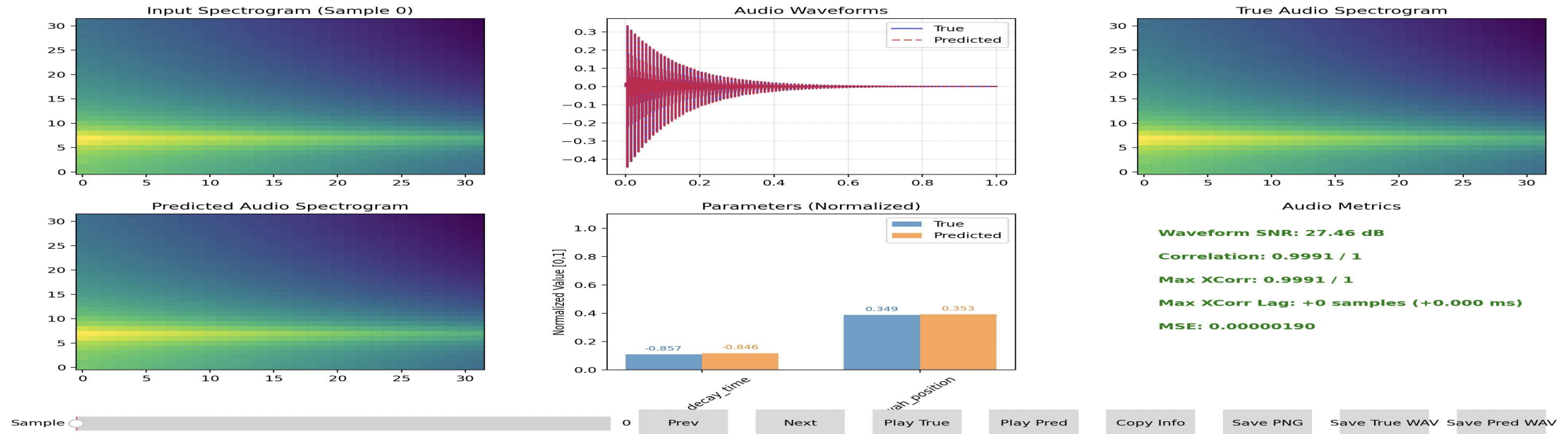


- *Self-describing*: Variable dimensions up to 65k×65k×65k (16-bit metadata)
- *Flexible labels*: 0-255 parameters, 8-bit quantized (100 steps)
- *Per sample*: [height, width, channels] + [N params] + [ID,val pairs] + image
- *Applications*: Audio spectrograms, vision, scientific data
- Models auto-configure from dataset metadata

Variable Image MultiHead (VIMH) Format

Generalized Dataset Format for Multi-Parameter Learning (Multi-Category Classification or Regression)

```
mode=regression | heads[log10_decay_time:1, wah_position:1] | weights: compat=20, skip=0, regression_loaded=yes  
run=2025-09-16_00-35-19 | dataset=saw_wah (32x32x1) | sr=8000 Hz, dur=1.0s | arch=SimpleCNN params=1,124,994 | steps=20600 epoch=199 | ckpt=logs/train/runs/2025-09-16_00-35-19/checkpoints/last.ckpt
```



- *Self-describing*: Variable dimensions up to 65k×65k×65k (16-bit metadata)
- *Flexible labels*: 0-255 parameters, 8-bit quantized (100 steps)
- *Per sample*: [height, width, channels] + [N params] + [ID,val pairs] + image
- *Applications*: Audio spectrograms, vision, scientific data
- Models auto-configure from dataset metadata
- Say make help | grep vimh



[Lightning Hydra Template *Ext*](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

LHTE Architectures: MLP += CNNs + ViT + ConvNeXt + EfficientNet





[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

- CNN
- YAML
- ConvNeXt
- EfficientNet
- ViT
- MultiHead

[Workflows](#)

SimpleCNN Architecture

SimpleCNN Architecture Flow



Convolutional Neural Network for Spatial Feature Extraction





[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

- CNN
- YAML
- ConvNeXt
- EfficientNet
- ViT
- MultiHead

[Workflows](#)

SimpleCNN Architecture

SimpleCNN Architecture Flow



Convolutional Neural Network for Spatial Feature Extraction

- MNIST and CIFAR-10/100 benchmark examples included, e.g.:





[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

- **CNN**
- YAML
- ConvNeXt
- EfficientNet
- ViT
- MultiHead

[Workflows](#)

SimpleCNN Architecture

SimpleCNN Architecture Flow



Convolutional Neural Network for Spatial Feature Extraction

- MNIST and CIFAR-10/100 benchmark examples included, e.g.:
 - *Two conv layers: 3×3 kernels, 32→64 channels, MaxPool, BatchNorm*





[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

- **CNN**
- YAML
- ConvNeXt
- EfficientNet
- ViT
- MultiHead

[Workflows](#)

SimpleCNN Architecture

SimpleCNN Architecture Flow



Convolutional Neural Network for Spatial Feature Extraction

- MNIST and CIFAR-10/100 benchmark examples included, e.g.:
 - *Two conv layers:* 3×3 kernels, 32→64 channels, MaxPool, BatchNorm
 - *Adaptive pooling:* 7×7 feature maps → FC classifier with dropout





[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

- CNN
- YAML
- ConvNeXt
- EfficientNet
- ViT
- MultiHead

[Workflows](#)

SimpleCNN Architecture

SimpleCNN Architecture Flow



Convolutional Neural Network for Spatial Feature Extraction

- MNIST and CIFAR-10/100 benchmark examples included, e.g.:
 - *Two conv layers:* 3x3 kernels, 32→64 channels, MaxPool, BatchNorm
 - *Adaptive pooling:* 7x7 feature maps → FC classifier with dropout
 - *Parameter variants:* 8K (small), 68K (medium), 421K (large):
`./configs/mnist_cnn_[8k|68k|421k].yaml`





[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

- **CNN**
- YAML
- ConvNeXt
- EfficientNet
- ViT
- MultiHead

[Workflows](#)

SimpleCNN Architecture

SimpleCNN Architecture Flow



Convolutional Neural Network for Spatial Feature Extraction

- MNIST and CIFAR-10/100 benchmark examples included, e.g.:
 - *Two conv layers:* 3×3 kernels, 32→64 channels, MaxPool, BatchNorm
 - *Adaptive pooling:* 7×7 feature maps → FC classifier with dropout
 - *Parameter variants:* 8K (small), 68K (medium), 421K (large):
./configs/mnist_cnn_[8k|68k|421k].yaml
 - *Performance:* CIFAR-10: 85-92%, CIFAR-100: 55-70%





[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

- CNN
- YAML
- ConvNeXt
- EfficientNet
- ViT
- MultiHead

[Workflows](#)

SimpleCNN Architecture

SimpleCNN Architecture Flow



Convolutional Neural Network for Spatial Feature Extraction

- MNIST and CIFAR-10/100 benchmark examples included, e.g.:
 - *Two conv layers:* 3×3 kernels, 32→64 channels, MaxPool, BatchNorm
 - *Adaptive pooling:* 7×7 feature maps → FC classifier with dropout
 - *Parameter variants:* 8K (small), 68K (medium), 421K (large):
./configs/mnist_cnn_[8k|68k|421k].yaml
 - *Performance:* CIFAR-10: 85-92%, CIFAR-100: 55-70%
- LHTE adds multihead classification for multi-task learning



LHTE Architecture Config Files

cifar10_cnn_64k.yaml	cifar100_efficientnet_210k.yaml	mnist_efficientnet_22k.yaml
cifar10_convnext_128k.yaml	cifar100_sdn_1m.yaml	mnist_efficientnet_7m.yaml
cifar10_convnext_210k.yaml	cifar100_vit_210k.yaml	mnist_mh_cnn_422k.yaml
cifar10_convnext_64k.yaml	cifar100mh_cnn_64k.yaml	mnist_sdn_68k.yaml
cifar10_efficientnet_210k.yaml	cifar100mh_convnext_210k.yaml	mnist_sdn_8k.yaml
cifar10_mh_cnn_64k.yaml	cifar100mh_efficientnet_210k.yaml	mnist_vit_210k.yaml
cifar10_vit_210k.yaml	cifar100mh_vit_210k.yaml	mnist_vit_38k.yaml
cifar100_cnn_1m_original.yaml	mnist_cnn_421k.yaml	mnist_vit_821k.yaml
cifar100_cnn_1m.yaml	mnist_cnn_68k.yaml	mnist_vit_995.yaml
cifar100_cnn_64k.yaml	mnist_cnn_8k.yaml	mnist_vit_pytorch.yaml
cifar100_coarse_cnn_64k.yaml	mnist_convnext_210k.yaml	multihead.yaml
cifar100_convnext_10m.yaml	mnist_convnext_68k.yaml	vimh_cnn_64k_ordinal.yaml
cifar100_convnext_1m.yaml	mnist_convnext_821k.yaml	vimh_cnn_64k_regression.yaml
cifar100_convnext_210k.yaml	mnist_convnext_8k.yaml	vimh_cnn_64k.yaml

LHT: example.yaml





[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

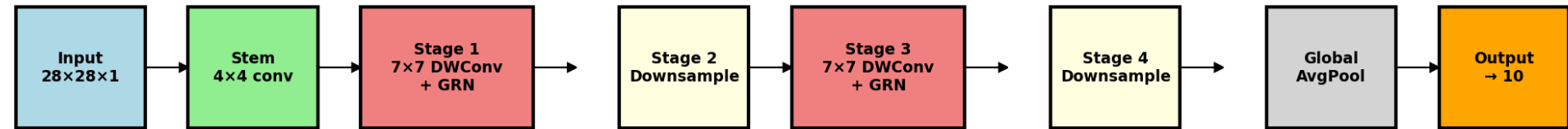
[New Architectures](#)

- CNN
- YAML
- **ConvNeXt**
- EfficientNet
- ViT
- MultiHead

[Workflows](#)

ConvNeXt-V2 Architecture

ConvNeXt-V2 Architecture with Global Response Normalization



Modern CNN with Global Response Normalization (GRN)

- *Key innovation:* GRN normalizes across spatial/channel dimensions
- *Architecture:* 7×7 depthwise conv, LayerNorm, 4× MLP with GELU
- *Sizes:* Tiny (18K), Small (73K), Base (288K), Large (725K)
- *Best performer:* CIFAR-10: 90-95%, CIFAR-100: 70-80%





[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

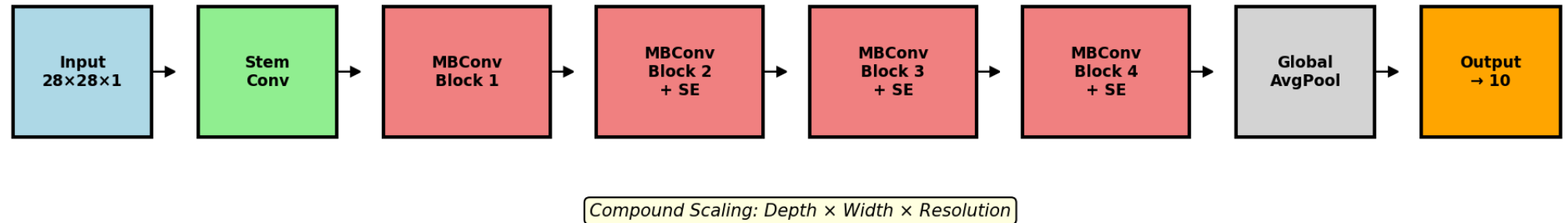
[New Architectures](#)

- CNN
- YAML
- ConvNeXt
- **EfficientNet**
- ViT
- MultiHead

[Workflows](#)

EfficientNet Architecture

EfficientNet Architecture with Mobile Optimization



Optimized CNN for Mobile and Edge Deployment

- *Compound scaling*: Balances depth, width, and resolution
- *Mobile-optimized*: Inverted residuals, squeeze-excitation, Swish
- *Sizes*: 22K (edge), 210K (balanced), 7M (high accuracy)
- *Performance*: CIFAR-10: 89-94%, CIFAR-100: 67-77%



[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

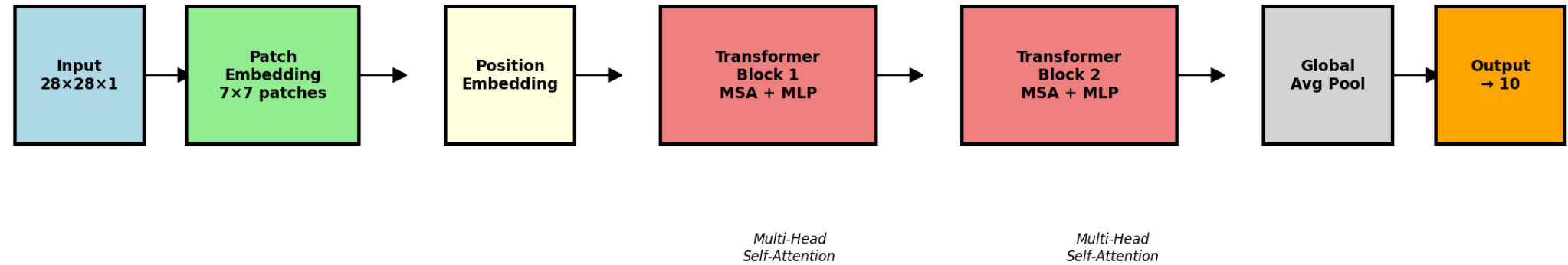
[New Architectures](#)

- CNN
- YAML
- ConvNeXt
- EfficientNet
- ViT
- MultiHead

[Workflows](#)

Vision Transformer (ViT) Architecture

Vision Transformer (ViT) Architecture



Attention-Based Learning on Image Patches

- *Image* → *Patches*: $28 \times 28 \rightarrow 7 \times 7$ patches → token sequence
- *Self-attention*: Multi-head attention + MLP with GELU
- *Sizes*: Tiny (38K), Small (210K, MNIST SOTA: 99.5%), Base (821K)
- Highly parallelizable, scales with data and compute



[Lightning Hydra Template Ext](#)

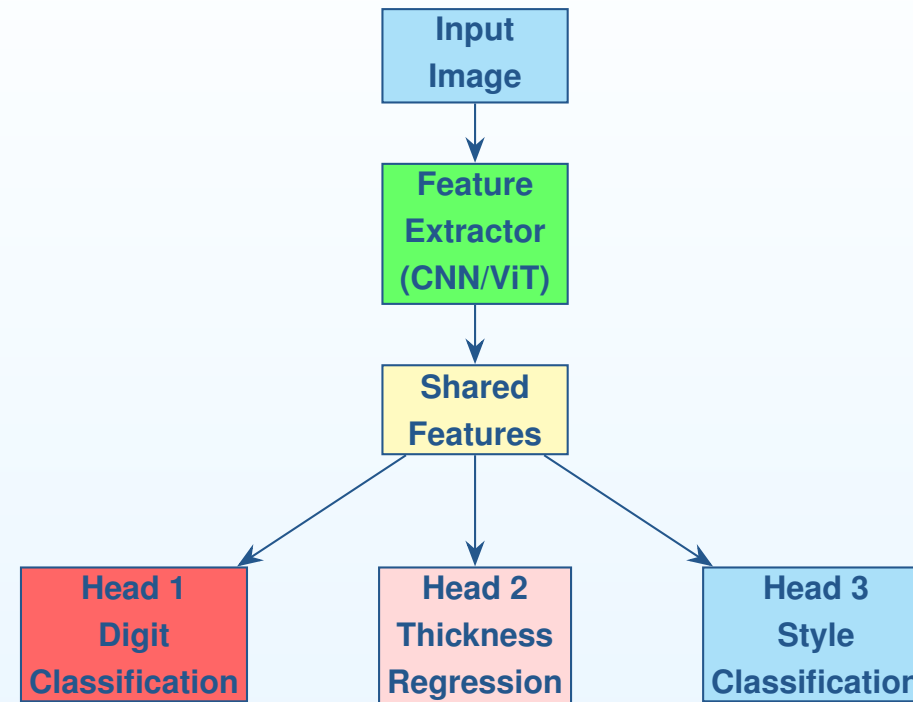
[New Dataset Formats](#)

[New Architectures](#)

- CNN
- YAML
- ConvNeXt
- EfficientNet
- ViT
- **MultiHead**

[Workflows](#)

Multi-Head Classification Support



Training Networks with Multiple Output Tasks



[Lightning Hydra Template Ext](#)

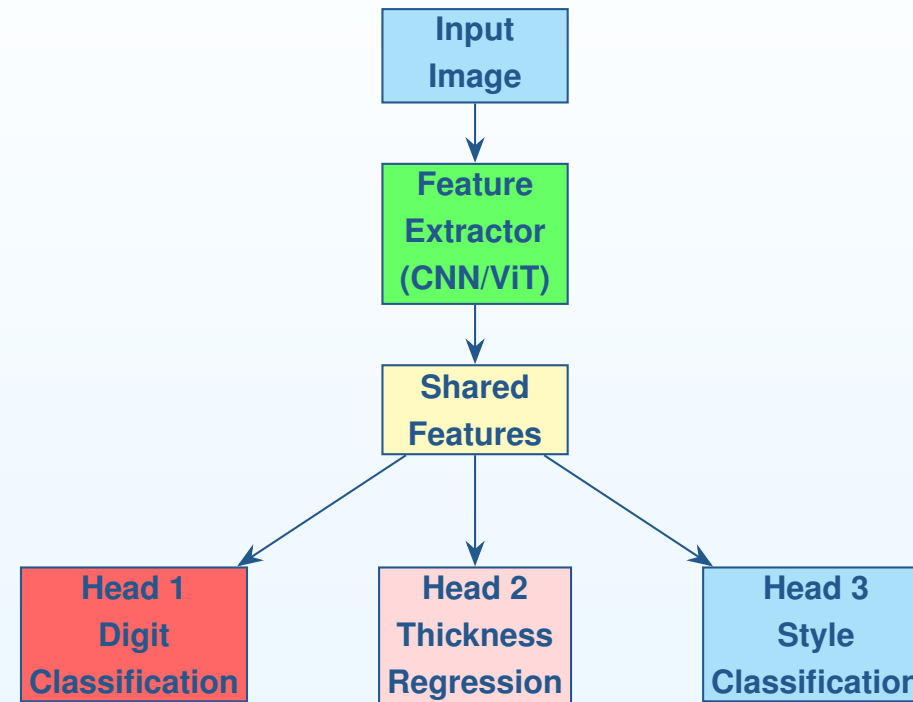
[New Dataset Formats](#)

[New Architectures](#)

- CNN
- YAML
- ConvNeXt
- EfficientNet
- ViT
- **MultiHead**

[Workflows](#)

Multi-Head Classification Support



Training Networks with Multiple Output Tasks

- *Architecture:* Shared backbone → separate heads per task





[Lightning Hydra Template Ext](#)

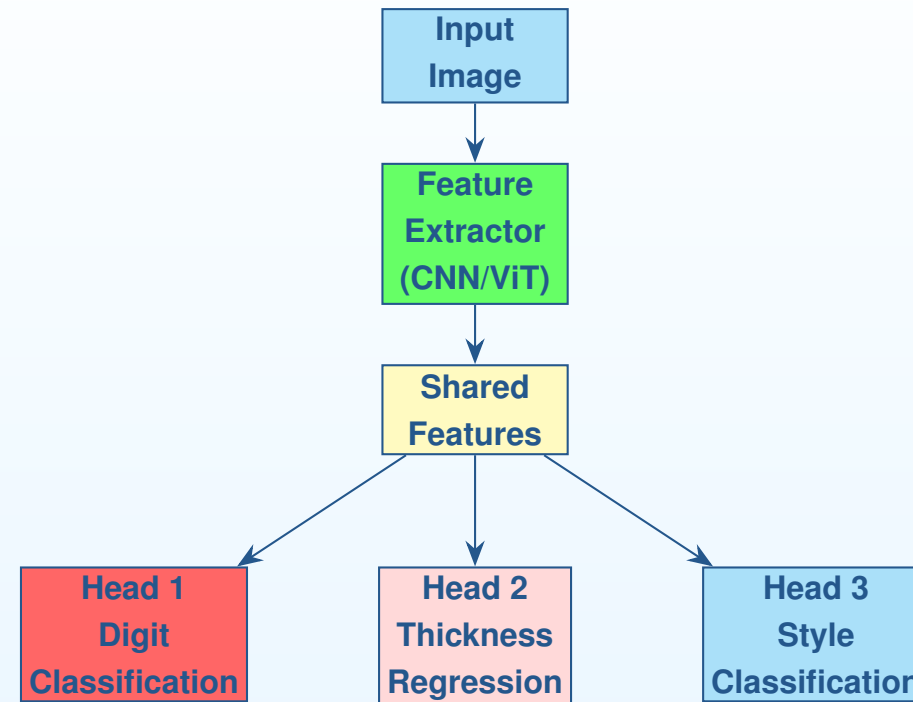
[New Dataset Formats](#)

[New Architectures](#)

- CNN
- YAML
- ConvNeXt
- EfficientNet
- ViT
- **MultiHead**

[Workflows](#)

Multi-Head Classification Support



Training Networks with Multiple Output Tasks

- *Architecture:* Shared backbone → separate heads per task
- *Task types:* Classification, ordinal regression, pure regression





Lightning Hydra Template Ext

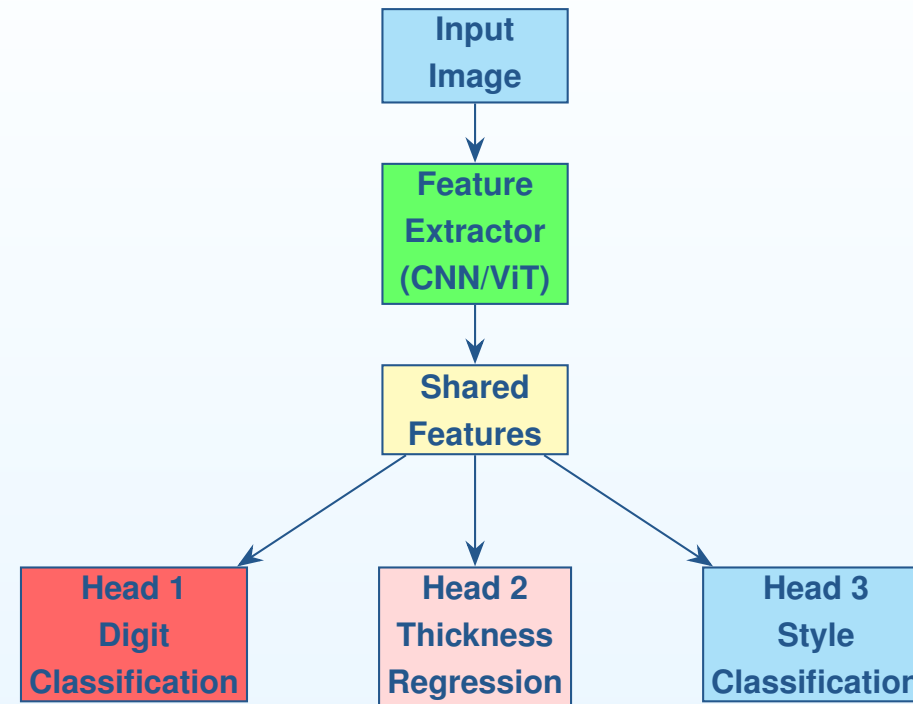
New Dataset Formats

New Architectures

- CNN
- YAML
- ConvNeXt
- EfficientNet
- ViT
- **MultiHead**

Workflows

Multi-Head Classification Support



Training Networks with Multiple Output Tasks

- *Architecture:* Shared backbone → separate heads per task
- *Task types:* Classification, ordinal regression, pure regression
- *Examples:* MNIST (digit+thickness), CIFAR-100 (fine+coarse), VIMH (synthesis-parameter + parameter-value)





Lightning Hydra Template Ext

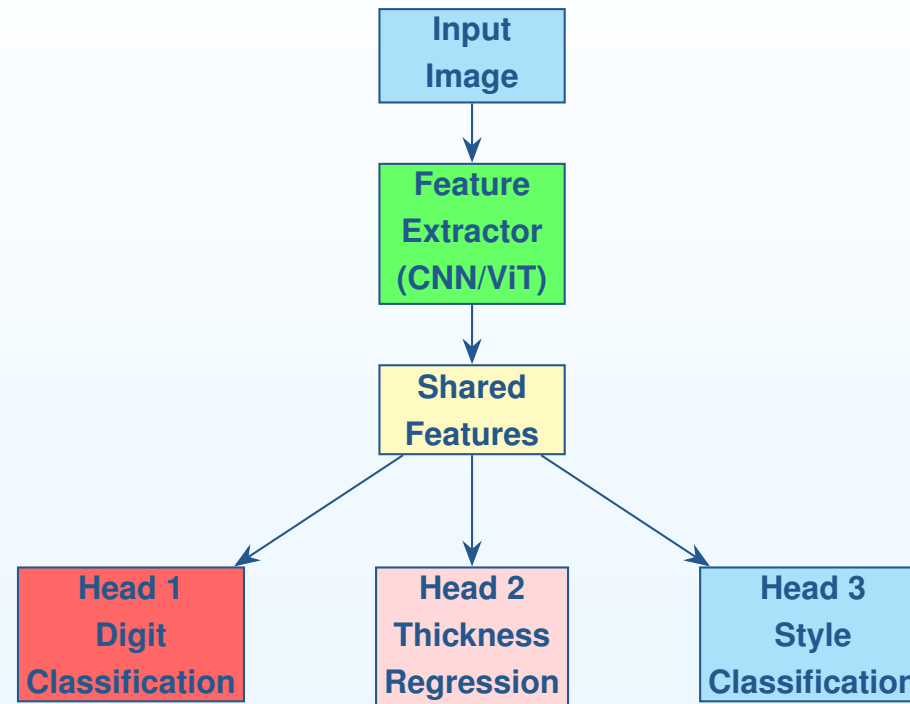
New Dataset Formats

New Architectures

- CNN
- YAML
- ConvNeXt
- EfficientNet
- ViT
- **MultiHead**

Workflows

Multi-Head Classification Support



Training Networks with Multiple Output Tasks

- *Architecture:* Shared backbone → separate heads per task
- *Task types:* Classification, ordinal regression, pure regression
- *Examples:* MNIST (digit+thickness), CIFAR-100 (fine+coarse), VIMH (synthesis-parameter + parameter-value)
- Configurable via Hydra without code changes





[Lightning Hydra Template *Ext*](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

Benchmarks, Experiments, and Future Work



[Lightning Hydra Template *Ext*](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

- **Benchmarks**
- Experiments
- Future Work

Benchmark Replications & Experiments

≈ 30 Pre-Configured Experiments



[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

- [Benchmarks](#)
- Experiments
- Future Work

Benchmark Replications & Experiments

≈ 30 *Pre-Configured Experiments*

- *Categories:* MNIST, CIFAR-10/100, VIMH, Multihead



[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

- [Benchmarks](#)
- [Experiments](#)
- [Future Work](#)

Benchmark Replications & Experiments

$\approx$ 30 *Pre-Configured Experiments*

- *Categories:* MNIST, CIFAR-10/100, VIMH, Multihead
- *Features:* Fixed seeds, fair comparisons, automated tracking





[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

- [Benchmarks](#)
- [Experiments](#)
- [Future Work](#)

Benchmark Replications & Experiments

≈ 30 Pre-Configured Experiments

- *Categories:* MNIST, CIFAR-10/100, VIMH, Multihead
- *Features:* Fixed seeds, fair comparisons, automated tracking
- *Example Make Commands* (abbreviated and full):



[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

- [Benchmarks](#)
- [Experiments](#)
- [Future Work](#)

Benchmark Replications & Experiments

≈ 30 Pre-Configured Experiments

- *Categories:* MNIST, CIFAR-10/100, VIMH, Multihead
- *Features:* Fixed seeds, fair comparisons, automated tracking
- *Example Make Commands* (abbreviated and full):
 - `make ca` (or `make compare-arch`)





[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

- **Benchmarks**
- Experiments
- Future Work

Benchmark Replications & Experiments

≈ 30 *Pre-Configured Experiments*

- *Categories:* MNIST, CIFAR-10/100, VIMH, Multihead
- *Features:* Fixed seeds, fair comparisons, automated tracking
- *Example Make Commands* (abbreviated and full):
 - `make ca` (or `make compare-arch`)
 - `make cbqa` (or `make cifar-benchmark-quick-all`)





[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

- [Benchmarks](#)
- [Experiments](#)
- [Future Work](#)

Benchmark Replications & Experiments

≈ 30 *Pre-Configured Experiments*

- *Categories:* MNIST, CIFAR-10/100, VIMH, Multihead
- *Features:* Fixed seeds, fair comparisons, automated tracking
- *Example Make Commands* (abbreviated and full):
 - `make ca` (or `make compare-arch`)
 - `make cbqa` (or `make cifar-benchmark-quick-all`)
 - `make cbsa` (or `make cifar-benchmark-suite-all`)





[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

- [Benchmarks](#)
- [Experiments](#)
- [Future Work](#)

Benchmark Replications & Experiments

≈ 30 *Pre-Configured Experiments*

- *Categories:* MNIST, CIFAR-10/100, VIMH, Multihead
- *Features:* Fixed seeds, fair comparisons, automated tracking
- *Example Make Commands* (abbreviated and full):
 - `make ca` (or `make compare-arch`)
 - `make cbqa` (or `make cifar-benchmark-quick-all`)
 - `make cbsa` (or `make cifar-benchmark-suite-all`)
- Version-controlled experiment configs ensure reproducibility



LHTE Experiment Config Files

cifar10_benchmark_cnn.yaml	cifar100_cnn.yaml	multihead_cnn_cifar10.yaml
cifar10_benchmark_convnext.yaml	cifar100_coarse_cnn.yaml	multihead_cnn_mnist.yaml
cifar10_benchmark_efficientnet.yaml	cifar100mh_cnn.yaml	vimh_cnn_16kdss_ordinal.yaml
cifar10_benchmark_vit.yaml	cifar100mh_convnext.yaml	vimh_cnn_16kdss_regression.yaml
cifar10_cnn_cpu.yaml	cifar100mh_efficientnet.yaml	vimh_cnn_16kdss.yaml
cifar10_cnn.yaml	cifar100mh_vit.yaml	vimh_cnn.yaml
cifar10_convnext_128k_optimized.yaml	cnn_mnist.yaml	vit_mnist_995.yaml
cifar10_convnext_64k_optimized.yaml	convnext_mnist.yaml	vit_mnist.yaml
cifar100_benchmark_cnn.yaml	convnext_v2_official...	
cifar100_benchmark_convnext.yaml	example.yaml	

LHT: mnist.yaml





[Lightning Hydra Template *Ext*](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

- Benchmarks
- Experiments
- **Future Work**

Future Work

Next Steps:

- Merge PRs
- Chatbot Reviews (GPT, Claude, Gemini, ...)



github.com/josmithiii/lightning-hydra-template-extended



[Lightning Hydra Template Ext](#)

[New Dataset Formats](#)

[New Architectures](#)

[Workflows](#)

- [Benchmarks](#)
- [Experiments](#)
- [Future Work](#)

Summary of Resources Online

- *Neural Spectral Modeling Template (NSMT):*
github.com/josmithiii/neural-spectral-modeling-template
- *Lightning Hydra Template Extended (LHTE):*
github.com/josmithiii/lightning-hydra-template-extended
- *Lightning Hydra Template (LHT):*
github.com/ashleve/lightning-hydra-template
- JOS Home Page (Videos, Overheads, including these):
<https://ccrma.stanford.edu/~jos/>

github.com/josmithiii/lightning-hydra-template-extended

