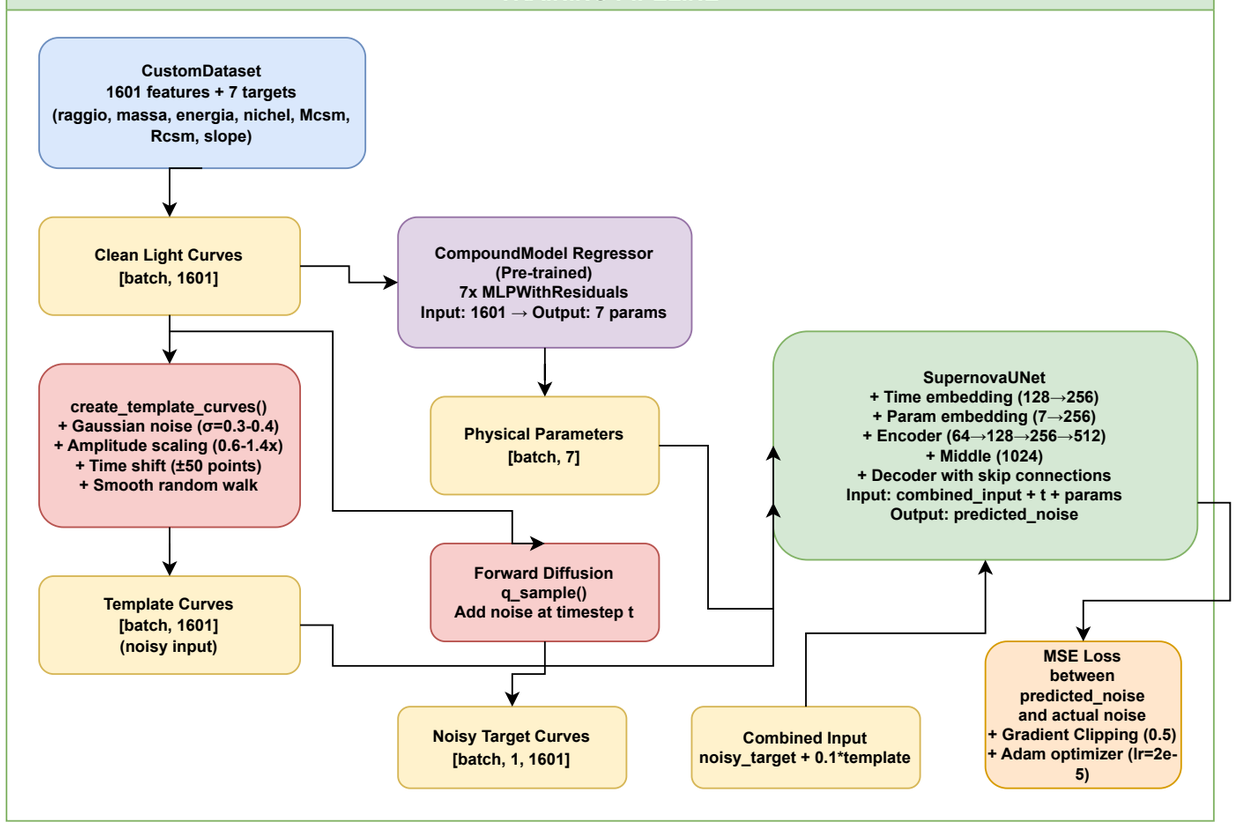
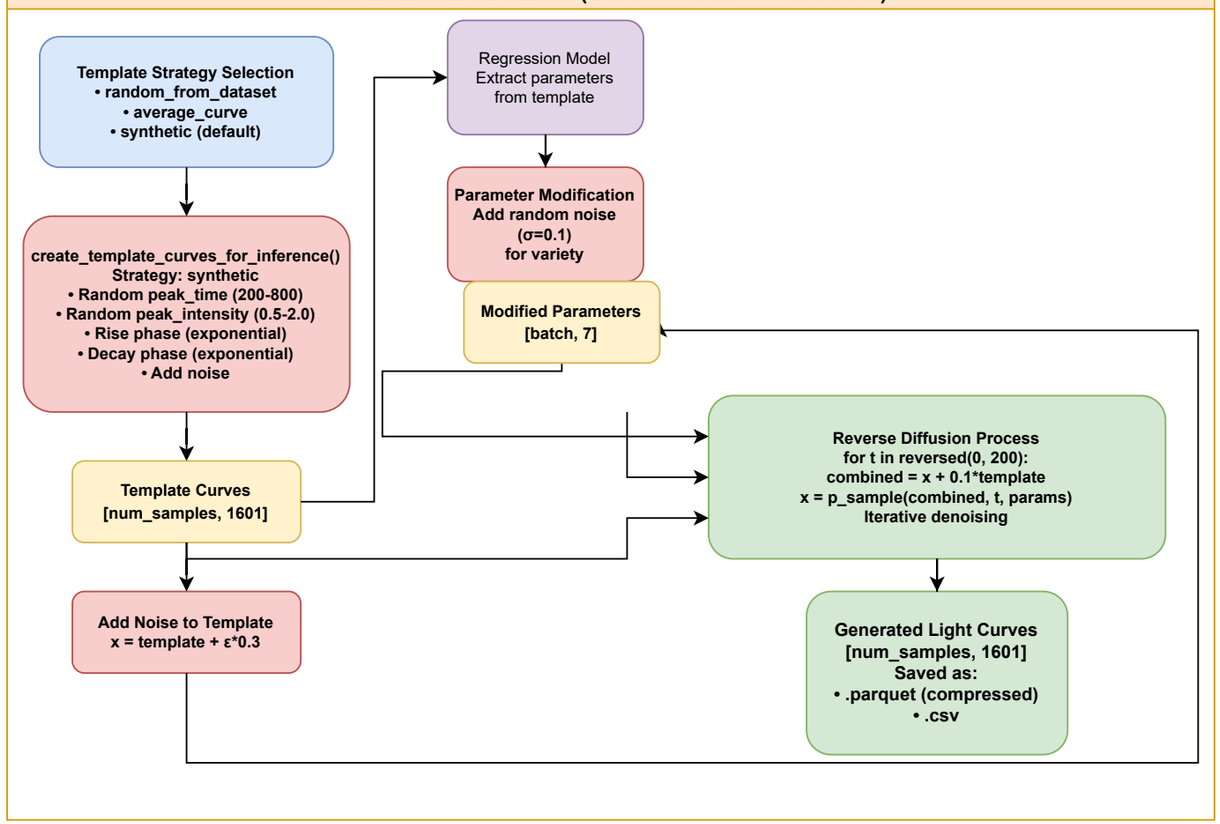
TRAINING PIPELINE



INFERENCE PIPELINE (Curve-to-Curve Generation)



KEY COMPONENTS

SupernovaUNet Architecture

Time Embedding: SinusoidalPositionEmbedding(128) → MLP(256)
Param Embedding: MLP(7→64→128→256)

Encoder:
Conv1d(1→64, k=7) → GroupNorm → SiLU
Conv1d(64→128, k=5, s=2) → GroupNorm → SiLU
Conv1d(128→256, k=5, s=2) → GroupNorm → SiLU
Conv1d(256→512, k=3, s=2) → GroupNorm → SiLU

Middle: Conv1d(512→1024→1024→512)

Decoder (with skip connections):
ConvTranspose1d + interpolation + concatenation
512→256→128→64→1

SupernovaDiffusion Process

Timesteps: 200
Beta schedule: Linear (0.0001 → 0.02)

Forward Process (q_sample):
$x_t = \sqrt{\bar{\alpha}_t} * x_0 + \sqrt{1-\bar{\alpha}_t} * \varepsilon$

Reverse Process (p_sample):
$x_{t-1} = \sqrt{1/\alpha_t} * (x_t - \beta_t*\varepsilon_\theta/\sqrt{1-\bar{\alpha}_t}) + \sqrt{\beta_t} * z$
where ε_θ is the predicted noise from UNet

Sampling:
Start from pure noise → iteratively denoise
Conditioned on physical parameters

CustomDataset

Features: 1601 time points (light curve)
Targets: 7 physical parameters
• raggio (radius)
• massa (mass)
• energia (energy)
• nichel (nickel)
• Mcsm (compact star mass)
• Rcsm (compact star radius)
• slope

Normalization: StandardScaler on features
Train/Val/Test split with shared scalers

Training Configuration

Batch size: 16
Learning rate: 2e-5
Optimizer: Adam (weight_decay=1e-5)
Scheduler: ReduceLROnPlateau
(factor=0.8, patience=15)
Gradient clipping: 0.5
Epochs: 600
Loss: MSE

Template noise level: 0.4
Amplitude scaling: 0.6-1.4x
Time shift: ±50 points
Combined input weight: 0.1

Output Files

Training:
• best_curve_to_curve_diffusion.pth
• last_curve_to_curve_diffusion.pth
• Best_mlp_res_for_diffusion.pth

Inference:
• generated_supernovae.parquet
• generated_supernovae.csv

Logging:
• Visdom (env: curve_to_curve_diffusion)

LEGEND

Data/Tensors | Processing/Transform | Neural Network | Pre-trained Model | Loss/Training | Input/Config

NOTES:
• Curve-to-Curve approach: starts from noisy template curves rather than pure noise
• Conditioning: physical parameters guide the generation process
• Skip connections in UNet preserve spatial information at multiple scales
• Template curves provide structural guidance, reducing generation variance
• Combined input (noisy + 0.1*template) maintains template influence during denoising