

NAM → NAN Technical White Paper

AI-Based Feature Extraction from NAM Model Structures and Weights

PREPARED BY: M-VAVE Audio Algorithms Team

VERSION: V1.0 — Pre-release

DATE : August 2026

CLASSIFICATION: Technical White Paper

Abstract — NAM → NAN converts models by directly parsing their structures and weights. It reads supported NAM FULL models and uses a cloud-based AI conversion model to extract mathematical features and generate NAN parameters, eliminating the need to generate training samples and retrain a target network for each source model. This paper describes the method’s technical principles, conversion workflow, and scope, and compares the two generations of conversion approaches with other common model migration methods.

1. Overview

NAM (Neural Amp Modeler) uses neural networks to model the nonlinear characteristics of amplifiers and distortion devices. On some platforms with limited processing capacity or no support for native NAM execution, a source model may need to be converted to a model supported by the target platform.

Terminology: In this paper, ANN refers to M-VAVE’s Audio Neural Network modeling technology; “neural network” is used for the general concept. NAN refers to the target model produced by this conversion approach and its deployment format. Support for NAM FULL must be defined by the specific architecture, model configuration, and file version; the name alone does not establish compatibility.

The released first-generation NAM → ANN approach uses training on audio samples: the source NAM model is run to generate input-output audio pairs, which are then used to train the target ANN model. Sample generation and target-model training are required for each source model. Conversion time and quality depend on sample coverage, training configuration, and convergence.

NAM → NAN uses the source model’s structure and weights as the basis for conversion. The new approach reads supported NAM FULL models directly and uses a cloud-based AI conversion model to extract mathematical features and generate NAN parameters. No audio samples are required during feature extraction or parameter generation.

First generation: NAM → Audio pairs → ANN model training → Validation

Next generation: NAM FULL structure and weights → Cloud-based AI feature extraction and parameter generation → NAN → Validation

2. Technical Principles

2.1 Cloud-Based AI Conversion Model

NAM models may differ in layer count, channel count, convolution configuration, and parameter count. Their weights must also meet the requirements of the NAN target architecture and cannot be used directly as NAN parameters. The conversion system uses a cloud-based AI conversion model to construct a representation for analysis from the input NAM model’s structure and weights and to perform the following tasks:

1.

Parse the structure
Read the NAM model’s file version, network topology, and weight tensors.
2.

Organize structural relationships
Identify the structural relationships among layers, channels, and convolution kernels.
3.

Extract features
Use AI to extract mathematical features from the model structure and weight tensors for NAN parameter generation.
4.

Build a unified representation
Organize relevant information from different NAM architectures into a unified intermediate representation.
5.

Generate parameters
Generate NAN model parameters from the extracted features and the NAN target architecture.

The representation used for analysis is constructed dynamically from the source NAM model’s topology and weights; “dynamically” describes how this representation is constructed. Feature extraction during conversion does not use audio samples, swept test signals, or direct instrument recordings (DI signals).

2.2 AI-Based Mathematical Feature Extraction

Training on audio samples learns the source model’s behavior from input-output relationships. The NAM → NAN approach analyzes the source model’s structure and parameters, including weight distributions, connections between layers, channel relationships, and convolution-kernel configurations, and extracts the features needed to generate NAN parameters. This analysis requires no external excitation signal.

This paper describes the workflow for an individual model conversion. “No per-model training” means that the target network does not need to be retrained for each source NAM model during conversion. The training method for the cloud-based AI conversion model itself is outside the scope of this paper.

3. Conversion Workflow

The workflow from source-model parsing to NAN parameter generation consists of the following three steps. These steps do not require running the source NAM model to generate training samples or retraining a target network for each source model.

1.

Parse NAM FULL
Read the network structure, weights, biases, activation-function configuration, and metadata, and verify that the model’s file version and network architecture are supported.
2.

Extract structural and weight features
The cloud-based AI conversion model organizes and analyzes the source model’s structure and weights, extracting features for NAN parameter generation, including connections between layers, weight distributions, channel relationships, and convolution-kernel configurations.
3.

Generate NAN
Generate target-model parameters from the extracted features and NAN runtime constraints, check parameter ranges, model format version, computational requirements, and memory requirements, and output a NAN model for deployment.

After model generation, independent audio testing is still required to assess conversion error, including objective error measurements and A/B listening comparisons. Tests should use consistent sample rates, input levels, gain settings, and time-alignment procedures.

4. Comparison of the Two Conversion Approaches

The table below describes this paper’s architectural modules and the engineering information that must be confirmed for each. It is not a list of implemented features.

Aspect	First-generation NAM → ANN (sample-based training)	Next-generation NAM → NAN (structure- and weight-based conversion)
Basis for conversion	NAM input-output audio pairs	Supported NAM FULL structures and weights
Run the source model to generate training samples?	Yes	No
Core operation	Train the target ANN model	Structural and weight feature extraction; NAN parameter generation
Intermediate data	Audio training dataset	AI-extracted mathematical features
Reproducibility	Affected by training randomness, configuration, and execution environment	Must be verified with fixed inputs, model versions, conversion configuration, and execution environment
Main workflow	Sample generation → Training and tuning → Validation	Parsing → Feature extraction → Parameter generation → Validation

4.1 Sample-Based Training (First Generation)

Advantages

- Relies primarily on the source model’s input-output behavior, with limited dependence on its internal structure
- Training samples can be generated using excitation signals if the source NAM model runs correctly
- Independent audio samples can be used to measure error between the target and source models
- Can serve as an alternative for models unsupported by weight-based conversion or for special cases

Limitations

- Each source model requires sample generation and training of a corresponding target model
- Conversion time depends on audio duration, computational resources, training epochs, and convergence criteria
- The target model may struggle to reproduce the source model’s behavior under input conditions not covered by the training samples
- Training configuration and random factors may cause conversion results to vary

4.2 Structure- and Weight-Based Conversion (Next Generation)

Advantages

- Reads supported NAM FULL structures and weights directly, without generating training samples
- Eliminates the need to retrain a target network for each source model, removing the corresponding training stage
- Weights, connections between layers, and channel relationships provide a direct basis for feature extraction and parameter generation
- NAN architectural and resource constraints can be considered together during parameter generation
- Removes random factors introduced by per-model training, helping to control the consistency of conversion results

Limitations

- Depends on converter support for the NAM file version, network architecture, and associated configuration
- Conversion quality depends on the cloud-based AI conversion model and the capabilities of its feature representation
- Unsupported structures and abnormal parameters must be detected, with clear rules for rejecting conversion or using an alternative approach
- The output targets the NAN architecture and deployment format; compatibility with other formats or runtimes requires separate validation
- Direct access to source-model information does not imply lossless conversion; independent audio testing is still required to assess error

The new approach removes training-sample generation and target-network training for each source model. Actual conversion time and computational resource consumption still depend on the source-model structure, cloud workload, and the versions of the converter and cloud-based AI model. Comparisons with the first-generation approach should use consistent test conditions.

Reproducibility must be verified with the same input file, converter version, cloud-based AI model weights and version, conversion configuration, and execution environment. The acceptance criterion must specify whether outputs must be identical or whether differences within a defined numerical tolerance are acceptable.

5. Comparison of NAM Model Migration Methods

The main differences among these methods are whether the source NAM model is run to generate audio samples, whether optimization is performed for each model, and whether the target model preserves the source-model structure. The table summarizes common implementations; the methods can also be combined.

Method	Core principle	Input information	Per-model processing	Target model characteristics	Main limitations
Native NAM execution	Implement the operations required by the source model, such as convolutions, activation functions, and state updates, on the target platform and execute the model directly	NAM structure and weights	No model conversion or retraining required	Preserves the source-model structure and parameters	The target platform must support the required operations; native execution itself does not reduce the parameter count, and actual resource usage depends on the implementation and hardware
IIR-Based Nonlinear System Identification	Run NAM to obtain audio samples, then fit its input-output relationship using structures such as Wiener-Hammerstein models built from IIR filters and static nonlinear blocks	Input-output audio pairs	Parameter fitting and optimization	Typically compact, which can help reduce DSP processing cost	Limited by the chosen structure and sample coverage; reproduction of nonlinear behavior with complex memory effects and harmonic characteristics under high-gain conditions requires separate assessment

Sample-Based Training of Lightweight Neural Networks	Use NAM as a teacher model to generate samples, then train a smaller student neural network by gradient descent	Input-output audio pairs	Training for each source model	Approximates the source model’s behavior with a smaller neural network	Requires training samples, computational resources, and optimization time; GPU acceleration can typically be used. Quality depends on sample coverage, loss function, and hyperparameters
--	---	--------------------------	--------------------------------	--	---

Direct Weight and Operator Mapping	Parse the NAM computational graph and map supported layers and weights to another inference format or runtime	NAM structure and weights	Typically requires no training	Usually preserves the source-network structure and parameters	The target runtime must support the required operators; the main purpose is format and execution compatibility, rather than reducing the parameter count
Pruning, Quantization, and Low-Rank Compression	Apply channel pruning, low-bit quantization, or low-rank approximation through matrix decomposition to a NAM model	NAM structure and weights; some methods also require calibration or fine-tuning data	Often requires calibration, fine-tuning, or error assessment	Can reduce model storage, memory traffic, or computation; benefits depend on the method and implementation	Requires balancing compression ratio, error, and hardware acceleration capabilities in the context of the source-model structure
NAM → NAN	The cloud-based AI conversion model reads supported NAM FULL structures and weights, extracts mathematical features, and generates NAN parameters	Supported NAM FULL structures and weights	Feature extraction and parameter generation; no per-model training	Generates model parameters for the NAN target architecture and runtime resource budget	Depends on supported NAM architectures, feature representation capabilities, and the cloud-based AI model version; conversion error requires independent validation

IIR-based nonlinear system identification and sample-based training of lightweight neural networks fit source-model behavior using input-output samples. Direct weight mapping usually preserves the source-network structure, while pruning, quantization, and low-rank compression modify the structure, numerical precision, or parameter representation. NAM → NAN uses a cloud-based AI conversion model to analyze the source model’s structure and weights, extract features, and generate NAN parameters.

6. Technical Scope and Release Information

NAM → NAN is currently pre-release technology. This paper does not yet include a complete compatibility list or measured performance results. The official release must specify the following:

- The architectures and model configurations covered by NAM FULL, along with supported file versions, sample rates, and associated metadata requirements
 - Version information for the converter, cloud-based AI conversion model, and feature representation
 - The NAN model format, runtime version, and supported target platforms
 - How unsupported models and abnormal parameters are detected, reported, and handled
 - Measured conversion times, objective audio errors, listening-test results, and runtime resource usage, together with the test models, hardware, sample rates, and conversion configurations
 - Reproducibility test conditions, results, and numerical tolerances
- Independent audio testing after conversion should compare the outputs of the source NAM and target NAN using identical inputs and consistent test conditions. Audio is not used for structural and weight feature extraction or NAN parameter generation during the conversion stages described above.

Key conclusion — NAM → NAN takes supported NAM FULL structures and weights as input and uses a cloud-based AI conversion model to extract mathematical features and generate NAN parameters, eliminating training-sample generation and target-network training for each source model.

This approach aims to simplify conversion from supported NAM FULL models to NAN. Conversion efficiency, audio fidelity, and runtime resource requirements should be evaluated for the actual models and target platforms. Its scope is determined by both converter support and validation results.

References
M-VAVE ANN White Paper: <https://yms-file-store.oss-cn-hongkong.aliyuncs.com/manual/white-paper/ANN.pdf>
Neural Amp Modeler official repository: <https://github.com/sdatkinson/neural-amp-modeler>
M-VAVE official download center: <https://www.m-vave.com/download>

This document is a pre-release technical description. “AI-based mathematical feature extraction” refers to the analysis of NAM structures and weights during model conversion. Compatibility and performance specifications are subject to official release information and measured results. This document does not constitute a commitment regarding specific conversion times or product performance. NAM and related names belong to their respective rights holders.

NAM → NAN 技术白皮书

基于 AI 的 NAM 模型结构与权重特征提取方法

编制：M-VAVE 音频算法团队

版本：V1.0 — 预发布

日期：2026 年 8 月

分类：技术白皮书

摘要 — NAM → NAN 采用直接解析模型结构与权重的转换路线：读取受支持的 NAM FULL 模型，通过云端 AI 转换模型提取数学特征并生成 NAN 参数，省去针对每个源模型生成训练样本和重新训练目标网络的环节。本文介绍该方法的技术原理、转换流程与适用边界，并对两代转换方案及其他常见模型迁移路线进行比较。

1. 概述

NAM（Neural Amp Modeler）使用神经网络对放大器和失真设备的非线性特征进行建模。在部分算力受限或不支持 NAM 原生运行的平台上，可能需要将源模型转换为目标平台支持的模型。

术语说明：本文中的 ANN 指 M-VAVE 的 Audio Neural Network 建模技术；泛指神经网络时，使用“神经网络”一词。NAN 指本转换方案输出的目标模型及其部署格式。NAM FULL 的支持范围需按具体架构、模型配置和文件版本界定，不能仅凭名称判断兼容性。

已发布的第一代 NAM → ANN 转换方案采用基于音频样本的训练路线：先运行源 NAM 模型，生成输入—输出音频样本对，再使用这些样本训练目标 ANN 模型。每个源模型都需要经历样本生成和目标模型训练，转换时间与质量受样本覆盖范围、训练配置及收敛情况影响。

NAM → NAN 采用基于源模型结构与权重的转换路线。新方案直接读取受支持的 NAM FULL 模型，通过云端 AI 转换模型提取数学特征并生成 NAN 参数。特征提取与参数生成阶段无需使用音频样本。

第一代：NAM → 音频样本对 → ANN 模型训练 → 验证
新一代：NAM FULL 结构与权重 → 云端 AI 特征提取与参数生成 → NAN → 验证

2. 技术原理

2.1 云端 AI 转换模型

不同 NAM 模型的层数、通道数、卷积配置和参数数量可能不同，其权重还需满足 NAN 目标结构的要求，不能直接作为 NAN 参数使用。转换系统通过云端 AI 转换模型，根据输入 NAM 模型的结构与权重构建用于分析的表示，并完成以下工作：

- 解析结构**
读取 NAM 模型的文件版本、网络拓扑与权重张量。
- 整理关系**
梳理各层、通道和卷积核之间的结构关系。
- 提取特征**
利用 AI 从模型结构与权重张量中提取生成 NAN 参数所需的数学特征。
- 统一表示**
将不同 NAM 结构的相关信息组织为统一的中间表示。
- 生成参数**
根据提取的特征和 NAN 目标结构生成 NAN 模型参数。

用于分析的表示根据源 NAM 模型的拓扑与权重动态构建；此处“动态”描述的是表示的构建方式。转换阶段的特征提取不使用音频样本、扫频信号或乐器直录（DI）信号。

2.2 基于 AI 的数学特征提取

基于音频样本的训练方法通过输入—输出关系学习源模型的行为。本方案则分析源模型的结构与参数，包括权重分布、层间连接关系、通道关系和卷积核配置等信息，并从中提取生成 NAN 参数所需的特征。该分析过程无需外部激励信号。

本文描述的是单次模型转换流程。“无需逐模型训练”指转换时无需针对每个源 NAM 模型重新训练目标网络；云端 AI 转换模型自身的训练方式不在本文讨论范围内。

3. 转换流程

从源模型解析到 NAN 参数生成，流程包含以下三个步骤。在这些步骤中，无需运行源 NAM 模型生成训练样本，也无需针对每个源模型重新训练目标网络。

- 解析 NAM FULL**
读取网络结构、权重、偏置、激活函数配置及元数据，检查模型的文件版本与网络结构是否受支持。
- 提取结构与权重特征**
云端 AI 转换模型整理并分析源模型的结构与权重，提取层间连接关系、权重分布、通道关系、卷积核配置等用于生成 NAN 参数的特征。
- 生成 NAN**
根据提取的特征与 NAN 运行约束生成目标模型参数，检查参数范围、模型格式版本、计算量及内存需求，输出用于部署的 NAN 模型。

模型生成后，仍需通过独立音频测试评估转换误差，包括客观误差计算和 A/B 听音比较。测试时应统一采样率、输入电平、增益与时间对齐条件。

4. 两代转换路线对比

下表给出本文的架构划分及各模块需要确认的工程信息，不作为已实现功能清单。

维度	第一代 NAM → ANN（样本训练）	新一代 NAM → NAN（结构与权重转换）
转换依据	NAM 输入—输出音频样本对	受支持的 NAM FULL 结构与权重
是否运行源模型生成训练样本	是	否
核心环节	训练目标 ANN 模型	结构与权重特征提取、NAN 参数生成
中间数据	音频训练数据集	AI 提取的数学特征
可复现性	受训练随机性、配置与执行环境影响	需固定输入、模型版本、转换配置与执行环境后验证
主要流程	样本生成 → 训练与调优 → 验证	解析 → 特征提取 → 参数生成 → 验证

4.1 样本训练路线（第一代）

优点 - 主要依赖源模型的输入—输出行为，对内部结构的依赖较少 - 源 NAM 模型能够正常运行时，可通过激励信号生成训练样本 - 可通过独立音频样本计算目标模型与源模型之间的误差 - 可作为权重转换不支持的模型或特殊场景的备选路线	限制 - 每个源模型都需要生成样本并训练对应的目标模型 - 转换时间受样本时长、计算资源、训练轮数和收敛条件影响 - 对训练样本未覆盖的输入条件，目标模型可能难以准确还原源模型的行为 - 训练配置与随机因素可能导致转换结果波动
---	--

4.2 结构与权重转换路线（新一代）

优点 - 直接读取受支持的 NAM FULL 结构与权重，无需生成训练样本 - 无需针对每个源模型重新训练目标网络，省去相应的训练环节 - 权重、层间连接和通道关系可作为特征提取与参数生成的直接依据 - 可在参数生成阶段统一考虑 NAN 的结构与资源约束 - 省去逐模型训练引入的随机因素，有利于控制转换结果的一致性	限制 - 依赖转换器对 NAM 文件版本、网络结构及相关配置的支持 - 转换质量受云端 AI 转换模型及其特征表示能力影响 - 需要检测不受支持的结构与异常参数，并明确拒绝转换或采用备选路线的处理规则 - 输出面向 NAN 结构与部署格式，与其他格式或运行时的兼容性需单独验证 - 直接读取源模型信息不代表转换结果无损，仍需通过独立音频测试评估误差
---	---

新方案省去针对每个源模型生成训练样本和训练目标网络的环节。实际转换耗时与计算资源消耗仍取决于源模型结构、云端任务负载、转换器及云端 AI 模型版本，应通过统一条件下的测试与第一代方案进行比较。

可复现性需在相同输入文件、转换器版本、云端 AI 模型权重与版本、转换配置及执行环境下验证，并说明判定标准是输出完全一致，还是差异处于规定的数值容限内。

5. NAM 模型迁移技术路线对比

不同路线的主要区别在于是否运行源 NAM 模型生成音频样本、是否针对每个模型进行优化，以及目标模型是否保留源模型结构。下表概括常见做法，各路线也可组合使用。

技术路线	核心原理	输入信息	单模型处理	目标模型特点	主要限制
NAM 原生运行	在目标平台实现源模型所需的卷积、激活函数及状态更新等运算，直接执行模型	NAM 结构与权重	无需模型转换或重新训练	保留源模型结构与参数	目标平台需支持相应运算；原生运行本身不减少模型参数规模，实际开销取决于实现与硬件
基于 IIR 的非线性系统辨识	运行 NAM 获取音频样本，以 IIR 滤波器与静态非线性模块构成 Wiener-Hammerstein 等结构，拟合其输入—输出关系	输入—输出音频样本对	参数拟合与优化	模型通常较紧凑，有利于降低 DSP 运行开销	受所选结构与样本覆盖范围限制；对具有复杂记忆效应的非线性行为及高增益条件下的谐波特征，还原能力需单独评估
基于样本的轻量神经网络训练	由 NAM 教师模型生成样本，通过梯度下降训练较小的学生神经网络	输入—输出音频样本对	针对每个源模型训练	以较小的神经网络近似源模型行为	需要训练样本、计算资源与优化时间，通常可使用 GPU 加速；质量受样本覆盖、损失函数和超参数影响
直接权重映射与算子映射	解析 NAM 计算图，将受支持的层与权重映射到另一推理格式或运行时	NAM 结构与权重	通常无需训练	通常保留源网络结构与参数	目标运行时需支持相应算子；主要解决格式与执行兼容问题，本身不以减少模型参数规模为目标
剪枝、量化与低秩压缩	对 NAM 模型进行通道剪枝、低比特量化或基于矩阵分解的低秩近似	NAM 结构与权重；部分方法还需校准或微调数据	常需校准、微调或误差评估	可降低模型存储量、访存量或计算量，收益取决于具体方法与实现	需结合源模型结构权衡压缩率、误差与硬件加速能力
NAM → NAN	云端 AI 转换模型读取受支持的 NAM FULL 结构与权重，提取数学特征并生成 NAN 参数	受支持的 NAM FULL 结构与权重	特征提取与参数生成；无需逐模型训练	根据 NAN 目标结构与运行资源预算生成模型参数	依赖受支持的 NAM 结构、特征表示能力及云端 AI 模型版本，转换误差需独立验证

基于 IIR 的非线性系统辨识与基于样本的轻量神经网络训练通过输入—输出样本拟合源模型行为。直接权重映射通常保留源网络结构；剪枝、量化与低秩压缩则对结构、数值精度或参数表示进行调整。NAM → NAN通过云端 AI 转换模型分析源模型结构与权重，提取特征并生成 NAN 参数。

6. 技术边界与发布说明

NAM → NAN 当前为预发布技术。本文尚未列出完整的兼容清单与性能实测结果，正式发布时需明确以下内容：

- NAM FULL 所对应的架构与模型配置，以及支持的文件版本、采样率和相关元数据要求
- 转换器、云端 AI 转换模型及特征表示的版本信息
- NAN 模型格式、运行时版本及适用的目标平台
- 不受支持的模型与异常参数的检测、提示及处理方式
- 转换耗时、客观音频误差、听音测试与运行资源占用的实测结果，并注明测试模型、硬件、采样率及转换配置
- 可复现性测试的条件、结果及数值容限

转换后的独立音频测试应比较源 NAM 与目标 NAN 在一致输入及测试条件下的输出。音频不参与上述转换阶段的结构与权重特征提取或 NAN 参数生成。

核心结论 — NAM → NAN 以受支持的 NAM FULL 结构与权重为输入，通过云端 AI 转换模型提取数学特征并生成 NAN 参数，省去逐模型的训练样本生成与目标网络训练。

该方案旨在简化受支持的 NAM FULL 到 NAN 的转换流程。其转换效率、音频保真度和运行资源需求应结合实际模型与目标平台进行评估；适用范围由转换器支持能力和验证结果共同确定。

参考资料

M-VAVE ANN 白皮书：<https://yms-file-store.oss-cn-hongkong.aliyuncs.com/manual/white-paper/ANN.pdf>

Neural Amp Modeler 官方代码库：<https://github.com/sdatkinson/neural-amp-modeler>

M-VAVE 官方下载中心：<https://www.m-vave.com/download>

本文为预发布技术说明。“基于 AI 的数学特征提取”指在模型转换阶段分析 NAM 的结构与权重。具体兼容范围与性能指标以正式发布说明和实测结果为准。本文不构成对具体转换时间或产品性能的承诺。NAM 及相关名称归其各自权利人所有。