Rapid Eye Analysis v2.0.28

Rapid Eye Analysis System

User Manual

October 07, 2026

A rapid eye diagram analysis system based on Python + PySide6 + scikit-rf

Table of Contents

Chapter 1 Software Overview 7

1.1 Key Features 7

1.2 Technology Stack 7

1.3 Use Cases 8

Chapter 2 System Requirements and Installation 9

2.1 System Requirements 9

2.2 Python Dependencies 9

2.3 One-click Installation (Recommended) 9

2.4 Manual Installation 10

Chapter 3 Quick Start 11

3.1 Your First Eye Diagram 11

3.2 Interface at a Glance 12

3.3 First Equalization: Enable CTLE 13

3.4 Bundled Demo Data (demo folder) 13

3.5 Where to Go Next 14

Chapter 4 Software Licensing 15

4.1 License Tiers 15

4.2 Viewing the Host ID 17

4.3 Activating a License 18

4.4 Deactivating a License 18

4.5 Status Bar License Information 18

4.6 Remote Activation via SCPI 18

4.7 Recovering an incorrect trial count 19

Chapter 5 Interface Overview 20

5.1 Menu Bar 20

5.2 Main Window — Eye Diagram Analysis 21

5.3 S-Parameter Analysis Windows 21

5.4 Language Switching 21

5.5 Help and About 21

Chapter 6 Format Conversion 23

6.1 Supported Conversions 23

6.2 Operation Steps 23

Chapter 7 Cascading and De-embedding 25

7.1 Multi-stage Cascading 25

7.2 De-embedding 26

Chapter 8 Splitting 28

8.1 Operation Steps 28

8.2 Smoothing 28

8.3 Accuracy Notes 29

Chapter 9 TDR and Smith Chart 30

9.1 TDR (Time-Domain Reflectometry) 30

9.2 Smith Chart (PLTS-aligned optimized) 31

9.3 Time-Domain Gating 32

Chapter 10 NRZ/PAM-N Virtual Eye Diagram 33

10.1 Basic Concepts 33

10.2 Signal Settings 34

10.3 Operation Steps 38

10.4 FFE Equalizer 39

10.5 RX Filter 39

10.6 CTLE Equalizer 40

10.7 RX FFE Equalizer 42

10.8 DFE Decision Feedback Equalizer 43

10.9 CDR (Clock Data Recovery) 45

10.10 Jitter Injection 48

10.11 Noise Injection 49

10.12 Eye Diagram Display Settings 50

10.13 Optimization Sampling Density 53

10.14 Optimization Algorithm Selection 53

10.15 GPU Acceleration 55

10.16 Streaming Processing 55

10.17 Crosstalk Injection 55

10.18 COM Import 57

Chapter 11 Expert Mode (Pipeline Designer) 60

11.1 Entering and Leaving 60

11.2 Canvas Operations 60

11.3 Node Types 61

11.4 Run and Results 62

11.5 Per-node Optimization 63

11.6 IBIS / IBIS-AMI Device Models 65

Chapter 12 Optical Link Simulation and TDECQ 71

12.1 Building an Optical Link 71

12.2 Optical Block Parameters 72

12.3 TDECQ Evaluation 73

Chapter 13 SCPI Remote Control 76

13.1 Starting the SCPI Server 76

13.2 Command Reference 76

13.3 S-Parameter Analysis Window Control 96

13.4 Eye Diagram Typical Workflow 100

Chapter 14 FAQ and Troubleshooting 102

Chapter 15 Pre- and Post-FEC BER Analysis 103

15.1 Enabling It 103

15.2 Standard Presets 104

15.3 Reading the Results 104

15.4 Patterns and Compute Time 105

15.5 Expert Mode 105

15.6 SCPI Control 106

15.7 Status Messages and Troubleshooting 107

15.8 MLSE Detector Measurement (measured gain) 107

Appendix A PLTS Alignment Overview 110

A.1 PLTS-aligned Defaults 110

A.2 TDR Output Formats (PLTS plot equivalents) 110

A.3 PLTS-equivalent Feature Map 111

A.4 Numerical Consistency 111

A.5 Remaining Differences 111

Appendix B Glossary 112

Chapter 1 Software Overview

Rapid Eye Analysis is a rapid eye diagram analysis system based on Python + PySide6, providing seven major functions: format conversion, cascading, de-embedding, splitting, TDR, Smith chart, and NRZ/PAM-N virtual eye diagram analysis.

1.1 Key Features

- **NRZ/PAM-N Virtual Eye Diagram**: eye simulation straight from channel S-parameters, full TX FFE / CTLE / RX FFE / DFE / CDR equalizer chain with auto-optimization, GPU accelerated
- **Expert Mode**: FlexDCA-style free signal-chain canvas with multi-probe eye comparison (see Chapter 11)
- **Optical Link Simulation**: EO / Fiber / OE blocks and TDECQ (IEEE 802.3 reference receiver) metric (see Chapter 12)
- **COM Import**: one-click import of IEEE 802.3 COM tool result CSV and configuration Excel
- **S-parameter Processing**: format conversion, multi-stage cascading, de-embedding, arbitrary-ratio splitting (defaults aligned with Keysight PLTS)
- **TDR and Smith Chart**: time-domain impedance analysis (incl. time-domain gating), interactive impedance trajectory
- **SCPI Remote Control**: 450+ command automation interface, bidirectionally synced with the GUI (see Chapter 13)
- **Bilingual**: instant Chinese / English UI switching with a manual and a quick start guide in both languages
- **Online updates and feedback**: check for new versions from the Help menu; type a message, paste screenshots and send them straight to the author - replies come back into the app (see 5.5)

1.2 Technology Stack

Component	Description
Python 3.14+	Runtime environment
PySide6	GUI framework (Qt 6)
scikit-rf	S-parameter reading and processing
NumPy / SciPy	Numerical computation, FFT, interpolation

Component	Description
Matplotlib	Chart rendering (embedded in Qt)
CuPy (optional)	GPU-accelerated eye diagram computation

1.3 Use Cases

- S-parameter post-processing for high-speed serial links (PCIe / USB / Ethernet)
- Signal integrity analysis for connectors, cables, and PCB channels
- Virtual eye diagram simulation: evaluate channel impact on NRZ/PAM4 signals
- Test fixture de-embedding and channel splitting
- TDR impedance profile analysis

Chapter 2 System Requirements and Installation

2.1 System Requirements

Item	Requirement
Operating System	Windows 10 / 11 (64-bit)
Python	3.14 or later (64-bit)
GPU Acceleration (optional)	NVIDIA GPU + CuPy (free, enabled by default, no license required)
Disk Space	Approx. 200 MB (excluding CuPy)

2.2 Python Dependencies

Package	Purpose
PySide6	GUI framework
scikit-rf	S-parameter reading and processing
numpy	Numerical computation
scipy	FFT, interpolation, optimization
matplotlib	Chart rendering
cupy (optional)	GPU acceleration

2.3 One-click Installation (Recommended)

The software includes a setup_env.bat one-click environment setup script that automatically detects, downloads, and installs Python 3.14 along with all runtime dependencies.

1. Double-click to run setup_env.bat.

   → GUI feedback: The script automatically detects Python 3.14 on the system (py launcher / PATH / common installation paths)

2. If Python 3.14 is not found, the script automatically downloads and installs it from the fastest mirror (Huawei Cloud / npmmirror / official) in passive mode with a progress bar.

3. Once Python is ready, the script automatically runs runfirst.py to install pip dependencies.

   → GUI feedback: runfirst.py opens a tkinter GUI, auto-detects the fastest PyPI mirror, and installs all dependencies

4. After installation, launch the software:

```
python main.py
```

> **Tip:** setup_env.bat requires no manual intervention. If Python is already installed and dependencies are ready, the script will quickly skip completed steps.

2.4 Manual Installation

To install manually, follow these steps:

1. Install Python 3.14+ (64-bit) and add it to the system PATH.
2. Navigate to the project directory and install dependencies:

```
pip install -r requirements.txt
```

3. Launch the software:

```
python main.py
```

4. (Optional) Install GPU acceleration support:

```
pip install "cupy-cuda12x[ctk]==14.0.1"
```

Chapter 3 Quick Start

This chapter walks you to your first eye diagram in about 5 minutes. All you need is a channel S-parameter file (.s4p or .s2p - a Touchstone file exported from a VNA or an EM simulator). Plain-language definitions of all terms are in Appendix B.

3.1 Your First Eye Diagram

Step 1: Launch. Double-click RapidEyeAnalysis.exe (or run python main.py from source). The main window is the eye-diagram workspace: parameter panel on the left, charts on the right.

Step 2: Load the channel. Click the "add file" button (folder icon) under the "Input Files" list at the top-left and pick your .s4p file. No file at hand? demo/sparams/demo_channel_clean.s4p in the install folder is a bundled demo channel (see 3.4).

> → GUI feedback: The file appears in the list; "File Info" below shows port count and frequency range.

Step 3: Confirm port pairing. A 4-port file needs to know which two ports form the differential pair. The default "Cross 1,3 | 2,4" fits the most common port definition (1/3 input pair, 2/4 output pair). Pick another pairing in the drop-down if your file differs. .s2p files skip this step.

Step 4 (optional): Signal settings. On the "Signal" page choose signal type (NRZ or PAM4), data rate (Gbaud) and pattern (default PRBS7). Defaults are fine for a first run.

Step 5: Generate. Click "Generate Eye" at the bottom of the left panel.

> → GUI feedback: The button shows progress; the eye appears on the right after a few seconds.

Step 6: Read the results. The cross markers on the eye show eye height (green vertical) and eye width (orange horizontal); the metrics bar under the chart lists eye height (mV), eye width (UI), COM (dB) and more.

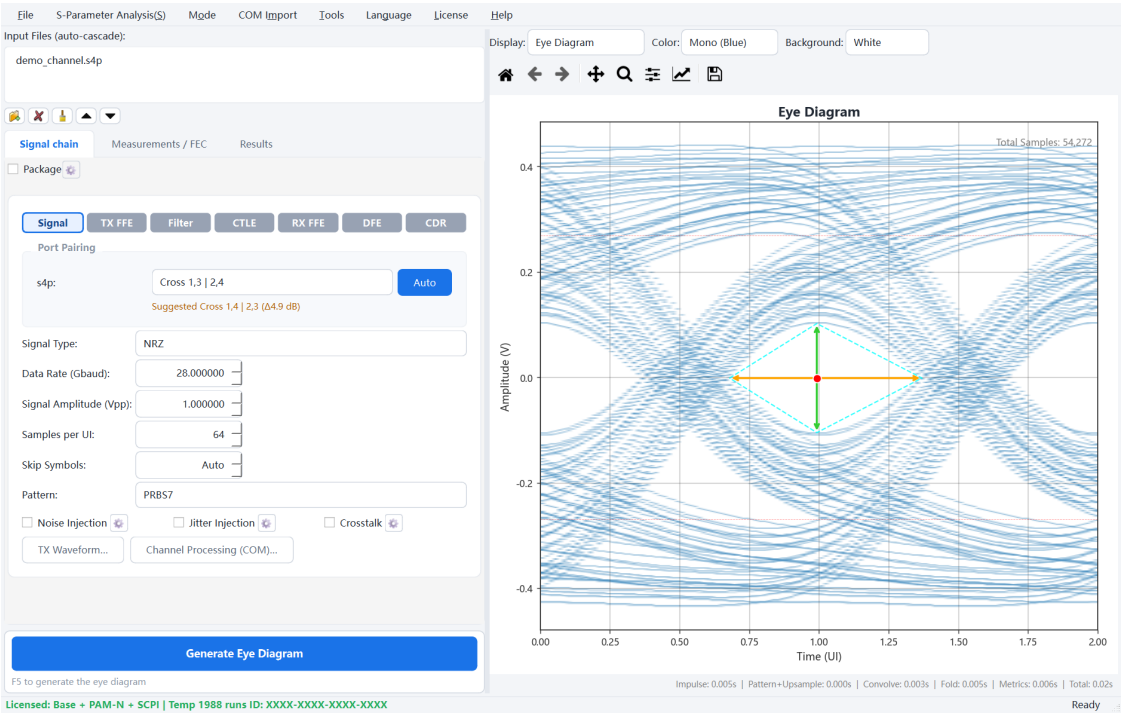

Figure 3-1 Main window after the first eye diagram

> **Tip:** Without a license the software runs in trial mode with 500 trial runs (NRZ and PAM-N eye generations share the counter; exports are metered at 10 per run). See Chapter 4 for activation.

3.2 Interface at a Glance

Area	Location	Purpose
Input file list	Top left	Load one or more S-parameter files (multiple files auto-cascade)
Function tabs	Left middle	Signal Settings / TX FFE / Filter / CTLE / RX FFE / DFE / CDR pages (order follows the signal chain; CDR sits after full equalization)
Generate Eye button	Bottom left	Compute and refresh the eye with current settings
Eye display	Right	Eye / impulse-response views and display options
Metrics bar	Bottom right	Eye height, eye width, COM, VEO and other results

Area	Location	Purpose
Status bar	Window bottom	License status, SCPI status, computation time

3.3 First Equalization: Enable CTLE

If the eye is closed (lossy channel), the quickest improvement is the receive-side Continuous Time Linear Equalizer (CTLE):

1. Open the "CTLE" tab and check "Enable CTLE".
2. Click "Optimize" - the software finds the best gain automatically.
3. Click "Generate Eye" again and watch eye height / width / COM improve.

> **Tip:** Every equalizer page (TX FFE / CTLE / RX FFE / DFE) has an "Optimize" button with the same meaning: automatically search that stage for the best parameters targeting COM (link margin).

3.4 Bundled Demo Data (demo folder)

The demo folder in the install directory ships ready-to-use data; every walkthrough in this manual can be reproduced with it:

Content	Purpose
sparams/demo_channel_clean.s4p	Clean channel - the first eye of the Quick Start (3.1)
sparams/lossy_channel_10dB.s4p	Lossy channel - reproduce the step-by-step equalization figures of 10.3-10.9
sparams/demo_trace.s2p	TDR / Smith chart / conversion demos
com_import/ (50G / 120G / 212G sets)	COM one-click import templates: CSV + Excel + channel (10.18)
configs/*.json	"File -> Load Configuration" samples: load the matching channel first, then the config, then Generate Eye
scpi_example.txt	SCPI command sequence to send line by line (Chapter 13)

> **Tip:** The configs do not embed channel paths: add the companion channel from sparams/ first (see README), load the config via "File -> Load Configuration", then click Generate Eye.

3.5 Where to Go Next

- Full eye-diagram and equalizer reference - Chapter 10
- S-parameter preprocessing (cascade / de-embed / TDR) - Chapters 6-9
- Free-form signal chains, multi-probe comparison - Chapter 11 Expert Mode
- Optical link simulation and TDECQ - Chapter 12
- Automation - Chapter 13 SCPI Remote Control

Chapter 4 Software Licensing

This software uses a Host ID-based licensing mechanism. On first launch, it runs in trial mode with 500 trial runs (NRZ and PAM-N eye generations share the counter; SCPI excluded). During the trial, S-parameter processing can be computed and viewed freely; exporting results is metered at 10 exports per trial run. Once the counter is exhausted, computing and viewing still work but exporting stops. For long-term use or advanced features, contact your vendor to obtain a license key.

One eye generation costs one run, in Normal and Expert mode alike. Optimize buttons **check the remaining balance before they start**; the eye that is generated automatically when optimization finishes is charged by the same rule. A COM result import costs exactly one run for the whole flow, even with the automatic optimizations enabled.

4.1 License Tiers

There are four key types. **PAM-N, SCPI and FEC are options: they only take effect on top of a permanent base key.** A temporary key only adds runs to the trial counter - it unlocks no permanent capability.

License Type	NRZ Eye	PAM-N / COM Import	SCPI	FEC	Runs
Trial (no key)	Yes (counted)	Yes (counted)	No	Yes (counted)	500
Permanent Base Key	Yes (free)	Yes (counted)	No	Yes (counted)	-
Base + PAM-N Option	Yes (free)	Yes (free)	No	Yes (counted)	-
Base + SCPI Option	Yes (free)	Yes (counted)	Yes	Yes (counted)	-
Base + FEC Option	Yes (free)	Yes (counted)	No	Yes (free)	-
Base + All Options	Yes (free)	Yes (free)	Yes	Yes (free)	-
Temporary Key (stacks)	per base tier	per base tier	not unlocked	per base tier	+N

Note: A temporary key **does not unlock SCPI**. SCPI execution commands require the permanent SCPI option; trial or temporary runs cannot substitute for it.

Tip: "Yes (counted)" means the feature works normally as long as runs remain: each generation consumes one run, and the feature is refused only when the counter reaches zero.

Tip: GPU / CUDA acceleration is free and enabled by default for all users; it needs no option license (only CUDA-capable hardware), so it is not listed separately in the license status and never consumes runs.

Tip: Temporary keys can coexist with permanent keys. Features covered by the base key (NRZ) do not consume temporary uses; features beyond the base key (PAM-N) consume temporary uses.

4.1.1 Per-Feature Licensing and Metering

The table below lists the rules the software actually enforces, feature by feature.

Feature	Requires	Metering
NRZ eye generation	Base key or runs > 0	1 run without a base key; free with one
PAM-N eye generation	PAM-N option or runs > 0	Free with the PAM-N option; otherwise 1 run
Expert mode pipeline run	Same (by chain signal type)	Same price as Normal: 1 run
Optimize buttons (CTLE / FFE / DFE / joint / TDECQ)	Same as eye generation, **checked before starting**	Optimization itself is free; the eye generated when it finishes is charged by the rule above
COM result import (incl. auto optimization)	PAM-N option or runs > 0	**Exactly 1 run for the whole flow**; free with the PAM-N option
S-parameter compute / view (cascade, de-embed, split, convert, TDR, Smith)	No restriction	Free

Feature	Requires	Metering
S-parameter result **export** (Save in those windows / SCPI SAVE)	Base key or runs > 0	**10 exports consume 1 run**; always free with a base key; cancelled or failed exports are not counted
SCPI execution commands (INIT / MEAS / OPT / COM import / pipeline run)	**SCPI option required**	Charged separately per the rules above
SCPI configuration / query / license commands	No restriction (server always available)	Free
Feedback / update check / manual / quick start	No restriction	Free

Tip: Running out of runs does not lock the software. S-parameters are still computed and displayed; only exporting result files stops, and eye generation resumes once a key is activated. Any valid key restores it immediately.

Note: With only a permanent base key, no PAM-N option and a zero counter, COM result import is unavailable - it needs either the PAM-N option or remaining (trial / temporary) runs.

4.1.2 What a Refusal Looks Like

- **GUI**: a license dialog appears with the WeChat QR code and this machine's Host ID; whatever you were editing is preserved.
- **SCPI**: the error queue returns **-203** (insufficient runs / license refused) or **-200** (missing SCPI option, missing PAM-N option). Query commands still return an empty string so the client never hangs.

4.2 Viewing the Host ID

1. Click the menu bar "License" → "License Management..."

   → GUI feedback: "Software License Management" dialog opens

2. The dialog displays this machine's Host ID (unique identifier). Click "Copy" to copy it to the clipboard.

3. Send the Host ID to your vendor to obtain the corresponding license key.

4.3 Activating a License

Permanent / Temporary License Key

In the "License Management" dialog, enter the key in the "Permanent / Temporary License Key" area and click "Activate". The software automatically identifies the key type:

- Paste the complete license key you received; the software identifies permanent or temporary licenses automatically.
- Do not identify the key type by its number of groups, shorten it, or edit it.

→ GUI feedback: Success: "Activation Successful" prompt appears

→ GUI feedback: Failure: "Activation Failed" warning appears

Option Key (PAM-N / SCPI / FEC)

In the "Option Key" area, enter the key and click "Activate". The software automatically identifies it as a PAM-N, SCPI or FEC option.

> **Note:** Option keys require an activated permanent base key first.

4.4 Deactivating a License

Click the "Deactivate" button in the corresponding area to deactivate a license. Deactivating the base license also clears the PAM-N, SCPI and FEC options.

4.5 Status Bar License Information

Status	Color	Description
Licensed	Green	Permanent key activated, features unrestricted
Temporary License	Blue	Temporary key activated, remaining uses shown
Trial Mode	Orange	Trial in progress, remaining uses shown

The lower-left corner of the status bar shows both the license status and Host ID, color-coded by license tier.

4.6 Remote Activation via SCPI

The SCPI server runs unconditionally at application startup. Even before the SCPI option is activated, you can remotely query the Host ID and activate license keys. This allows batch deployment or unattended machines to be authorized without opening the GUI.

Common commands:

- **LIC:HOST?** — Retrieve Host ID
- **LIC:ACT:BASE "key"** — Activate the base key
- **LIC:ACT:PAM/SCPI/TEMP "key"** — Activate corresponding option or temporary key
- **LIC:STAT?** — Query the full license status
- **LIC:ACT:LAST?** — View detailed result of the last activation

See Chapter 13 SCPI Remote Control for the full command list.

4.7 Recovering an incorrect trial count

In rare cases the trial count may read as exhausted after a version upgrade even though far fewer runs were actually used. This comes from how earlier versions identified the host hardware and has **nothing to do with your actual usage**. This release corrects the identification logic and can return the affected runs once.

On the first start after upgrading, the software checks whether the count was affected:

- **Confirmed affected** - the count is restored to the real number of runs used, with nothing for you to do; the status bar shows the restored balance immediately.
- **Cannot be confirmed automatically** - the licence dialog (**Help - Licence Management**) shows a short notice and an **Apply** button. It opens the feedback window with the required details **already filled in**; review them and send it yourself, and a temporary licence key will be issued once verified.

If the runs were genuinely used up, nothing is changed - the check targets only the fault described above, and it runs **only once**.

> **Tip:** Machines with an activated permanent base key are unaffected and never see this notice.

Chapter 5 Interface Overview

The main window is centered on eye diagram analysis. S-parameter analysis functions are accessed via the menu bar, each opening in a separate window.

5.1 Menu Bar

Menus from left to right:

Menu	Functions
File	Save/load configuration, recent files, optimization algorithm (search method, step, COM Top-N, optimization sampling density 16/32/64 pts/UI, hardware resources), initialize, exit
S-Param Analysis	Format conversion, cascading, de-embedding, splitting, TDR, Smith chart (each opens a separate window, Ctrl+1..6)
Mode	Normal / Expert working mode switch and the "Copy Normal Config to Expert" button (see Chapter 11)
COM Import	One-click COM result import dialog (see 10.18)
Tools	Enable CUDA acceleration, CUDA acceleration status, SCPI remote control toggle and status
Language	Instant Chinese / English UI switch
License	License management (activate / deactivate / Host ID)
Help	Quick start guide, user manual (CN/EN, F1), open log folder, send feedback, check for updates, about

5.1.1 Initialize (Reset)

File → Initialize performs two actions simultaneously:

- Clears input files in all S-parameter analysis windows
- **Immediately resets all controls to PLTS-aligned defaults** (no program restart required)

All spinboxes/comboboxes/checkboxes in the 6 S-parameter windows (Format Conversion, Cascade, De-embed, Split, TDR, Smith Chart) instantly switch to PLTS defaults. For example:

- TDR: window=plts_nominal, velocity_factor=0.7, dc_method=linear, causality=OFF
- Smith Chart: chart_type=Z, Z_ref=50Ω, VSWR circles=ON
- Cascade/Deembed/Splitter: port mapping=Adjacent (1,3 in / 2,4 out)

5.2 Main Window — Eye Diagram Analysis

The main window is the NRZ/PAM-N virtual eye diagram analysis interface, featuring a left control panel and a right chart display area. The left panel provides file selection, signal parameter configuration, FFE settings, and other controls; the right side displays the eye diagram histogram, metrics, and channel frequency-domain curves.

5.3 S-Parameter Analysis Windows

S-parameter analysis windows can be opened via the "S-Param Analysis" menu (multiple windows can be open simultaneously). Each window has a left control panel and a right chart display area. Keyboard shortcuts Ctrl+1 through Ctrl+6 provide quick access to each function.

Shortcut	Function
Ctrl+1	Format Conversion
Ctrl+2	Cascading
Ctrl+3	De-embedding
Ctrl+4	Splitting
Ctrl+5	TDR
Ctrl+6	Smith Chart

5.4 Language Switching

Rapid Eye Analysis supports bilingual operation in Chinese and English. The language can be switched at any time via the "Language" menu in the menu bar.

- **Menu Path**: Menu bar → "Language" → select "■■" (Chinese) or "English"
- **Instant Switch**: All UI text (menus, labels, buttons, tooltips) switches to the selected language immediately without restarting.
- **Preference Persistence**: The selected language is saved automatically and persists across application restarts.
- **Manual Language**: The user manual is also available in both languages. Access via "Help" → "User Manual" to open the manual in the current UI language.

5.5 Help and About

The "Help" menu provides: the quick start guide (9 steps, written for first-time trial users), the user manual (CN/EN PDF, F1), "Open Log Folder" (the diagnostic log directory - zip and send to support when reporting issues), **"Send Feedback"** (type a message, paste screenshots with Ctrl+V and it goes straight to the author; replies come back as an unread badge on the menu

entry), "Check for Updates", and "About" (version, copyright, third-party open-source licenses).

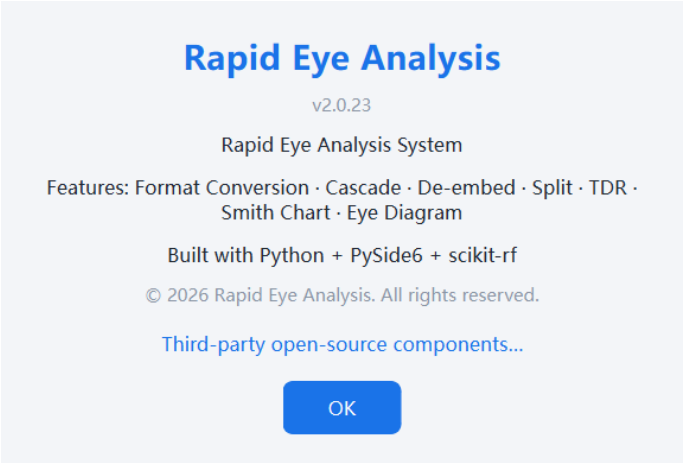

Figure 5-1 About dialog

Chapter 6 Format Conversion

The format conversion function supports various conversion operations on S-parameter files. Access it via menu bar "S-Param Analysis" → "Format Conversion" or shortcut Ctrl+1.

6.1 Supported Conversions

- **Data Format Conversion**: Convert between MA (magnitude/angle), DB (decibel/angle), and RI (real/imaginary) formats
- **Frequency Unit Conversion**: Convert between Hz, kHz, MHz, and GHz
- **Port Extraction**: Extract a specified port subset from multi-port S-parameter files
- **S4P Differential Mixed-Mode Conversion**: Convert 4-port single-ended S-parameters to differential/common-mode mixed-mode parameters (SDD, SDC, SCD, SCC)
- **Port Order Adjustment**: Remap port numbering to match different VNA conventions, useful when port order differs between instruments

6.2 Operation Steps

1. Select the source S-parameter file in the left panel (supports .s1p through .s4p Touchstone formats).
2. Select the target format and parameters.
3. Click the "Convert" button to execute the conversion.
4. Preview the result in the right chart area. When satisfied, click "Save" to export the converted file.

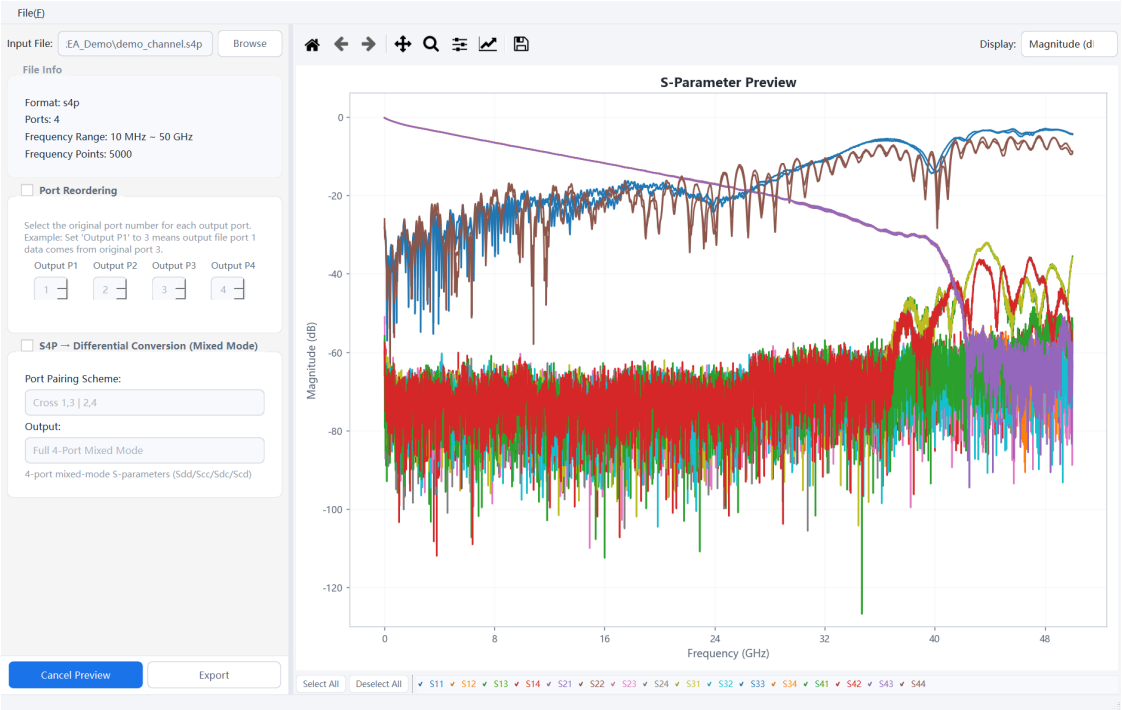

Figure 6-1 Format Conversion — S4P Differential Mixed-Mode

Chapter 7 Cascading and De-embedding

7.1 Multi-stage Cascading

The cascading function uses Scattering Transfer Matrix (T-matrix) chain multiplication to concatenate multiple network segments. Automatic frequency interpolation alignment is supported.

1. Add the S-parameter files to be cascaded (in connection order).

2. Use the up/down buttons to adjust the cascading order.

3. For 4-port networks, enable "Port Order" to specify input-side and output-side port numbers (default: input P1,P3 / output P2,P4).

4. Click the "Cascade" button to execute the computation.

5. The right panel displays the frequency-domain curves of the cascaded result, which can be saved as a new S-parameter file.

> **Tip:** Cascading computation has extremely high precision (~1e-16) and introduces negligible numerical error. When frequency points differ across files, the software automatically uses cubic spline interpolation for frequency alignment.

7.1.1 Mixed s2p / s3p / s4p Cascade

Both the cascade tool and the eye diagram "Input Files" list support mixing single-ended and differential S-parameters of any type in any order: s4p (differential 4-port) extracts differential Sdd by port pairing, s3p (single-ended ↔ differential) is reduced to [single-ended, differential] by its differential pair, and s2p (single-ended 2-port) is cascaded in list order in the 2-port differential domain, yielding a 2-port network. In the cascade tool, the s2p segment direction is set by "s2p Ports (In/Out)" and the s4p segment by the Port Order adjustment.

Figure 7-1 Cascade — Two S4P segments with differential conversion

7.2 De-embedding

De-embedding removes the influence of known test fixtures or adapters from the device under test (DUT). Three modes are supported: left-side, right-side, and dual-side de-embedding.

Mode	Description	Mathematical Principle
Left-side	Remove input-side (TX) fixture	T_result = T_fixture■¹ · T_DUT
Right-side	Remove output-side (RX) fixture	T_result = T_DUT · T_fixture■¹
Dual-side	Remove fixtures from both sides	T_result = T_L■¹ · T_DUT · T_R■¹

1. Select the S-parameter file for the DUT (including fixtures).

2. Select the fixture S-parameter file(s) (left / right / both).

3. For 4-port networks, enable "Port Order" to specify input-side and output-side port numbers. This setting applies to both DUT and fixture files.

4. Choose the de-embedding mode and click "De-embed".

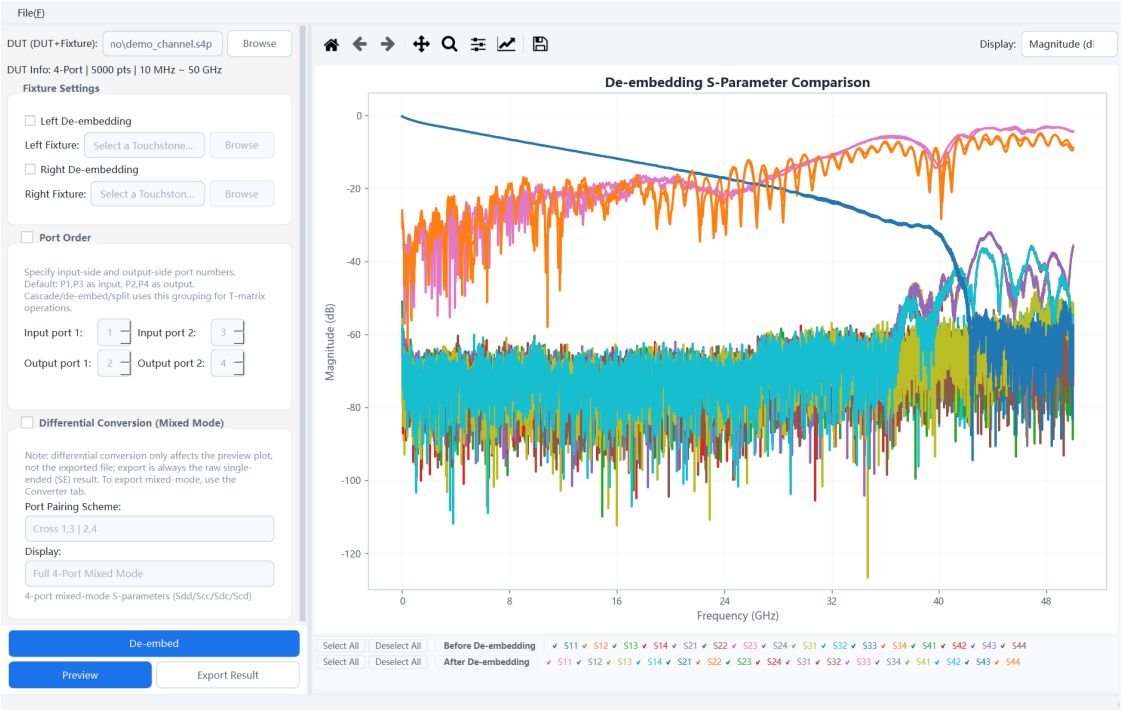

Figure 7-2 De-embedding — Dual-side de-embed with differential conversion

Chapter 8 Splitting

The splitting function divides a long channel into two segments at an arbitrary ratio. 2-port uses ABCD matrix fractional power; 4-port uses WCM T-matrix fractional power with eigenvalue tracking.

8.1 Operation Steps

1. Select the S-parameter file to split.

2. For 4-port networks, enable "Port Order" to specify input-side and output-side port numbers (default: input P1,P3 / output P2,P4).

3. Set the number of segments (2–10) using the "Number of Segments" spin box. The ratio input fields update dynamically to match.

4. Set the split ratio for each segment (e.g., 0.5 / 0.5 for equal halves). Ratios are automatically normalized.

5. Click the "Split" button to execute the computation.

6. The right panel displays frequency-domain curves of the resulting segments, each of which can be saved separately.

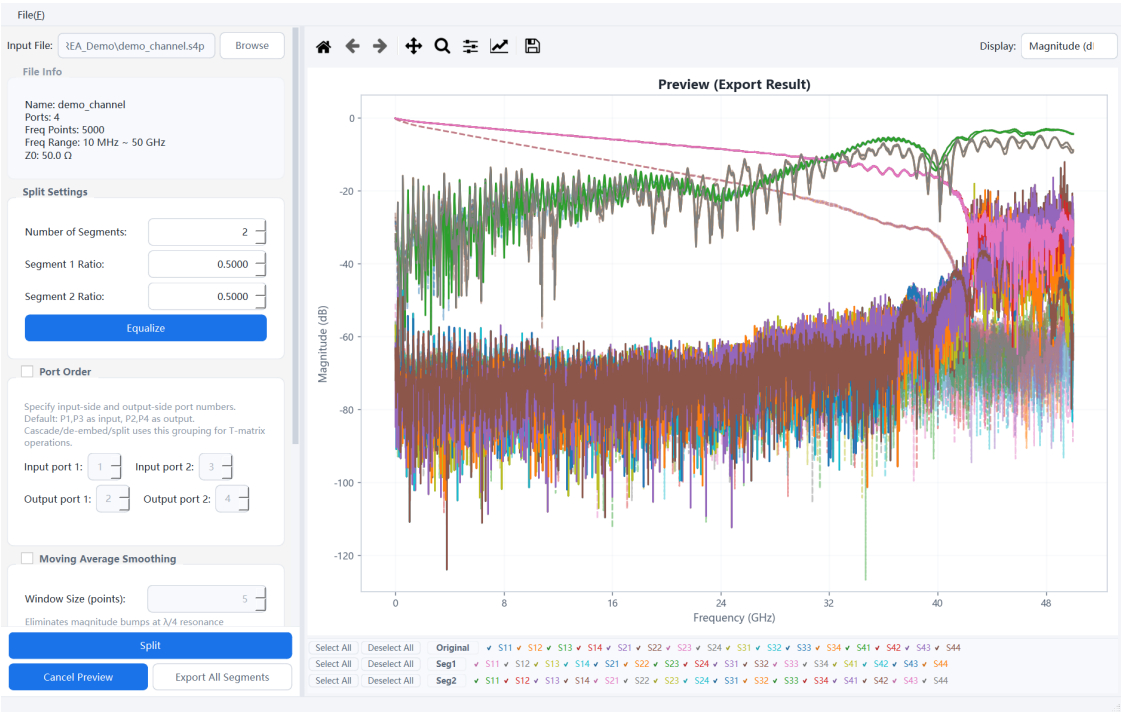

Figure 8-1 Splitting — 4-port S-parameter equal split with differential conversion

8.2 Smoothing

An optional post-split smoothing filter is available to reduce high-frequency noise artifacts. Enable the "Smoothing" checkbox to activate it.

• **Window Size**: Set the moving-average window size (odd number, range 3–51, default 5). Larger values produce smoother curves but may attenuate sharp features.

Tip: Smoothing is applied only for display and export; it does not modify the underlying split computation.

8.3 Accuracy Notes

Port Count	Algorithm	Reconstruction Accuracy
2-port	ABCD matrix fractional power	~2e-2
4-port	WCM T-matrix + Hungarian algorithm eigenvalue tracking	~4e-13 (machine precision)

Note: For 4-port splitting, amplitude bumps of <2% may occur at frequencies where eigenvalue phases pass through odd multiples of π. This is a physical $\lambda/4$ resonance characteristic, not an algorithm defect.

Chapter 9 TDR and Smith Chart

9.1 TDR (Time-Domain Reflectometry)

TDR analysis performs an IFFT on the S11 parameter to obtain the time-domain impedance profile, useful for locating impedance discontinuities in the channel. All defaults are PLTS-aligned, providing PLTS-equivalent behavior out of the box.

Thirteen window functions: plts_nominal (default, PLTS Nominal), plts_flat, plts_fast, exponential_plts, gaussian, kaiser, hamming, hanning, blackman, chebyshev, plts_custom, exponential, rectangular.

Click the "Window Function Settings" button to open a PLTS-compatible configuration dialog with 4 PLTS presets: Flat Response (TD 0.80), Nominal (TD 0.50, DEFAULT), Fast Rise Time (TD 0.20), and Custom, with real-time display of Step Rise Time and Impulse Width.

Display Mode: 6 PLTS-equivalent formats — Impedance (Ω, default STEP_IMPEDANCE_NS), Voltage (V, STEP_VOLTS_NS), Step (Γ, STEP_REAL_NS), LogMag (dB, STEP_LOGMAG_NS), Impulse (IMPULSE_REAL_NS), Impulse LogMag (IMPULSE_LOGMAG_NS).

DC Extrapolation: 5 methods — Auto (linear, PLTS default), Auto with PNA Method (circle), Even-Poly (REA legacy), First Re(S11) (SE T11 alignment), Manual (user-specified).

Advanced PLTS-equivalent tools: Correct Impedance Profile (Goldberg layer-stripping), One-way Display, Bandwidth Extension (AR spectral), Step Smoothing σ, Frequency Band Limiting, Manual DC value.

When a 4-port S-parameter file is loaded, the software automatically detects differential mode (Sdd) and sets the reference impedance to Z0=100Ω.

1. Select an S-parameter file.
2. Choose the window function type and configure parameters (or use a preset).
3. Click the "Compute TDR" button.
4. The right panel displays the impedance vs. time/distance curve.

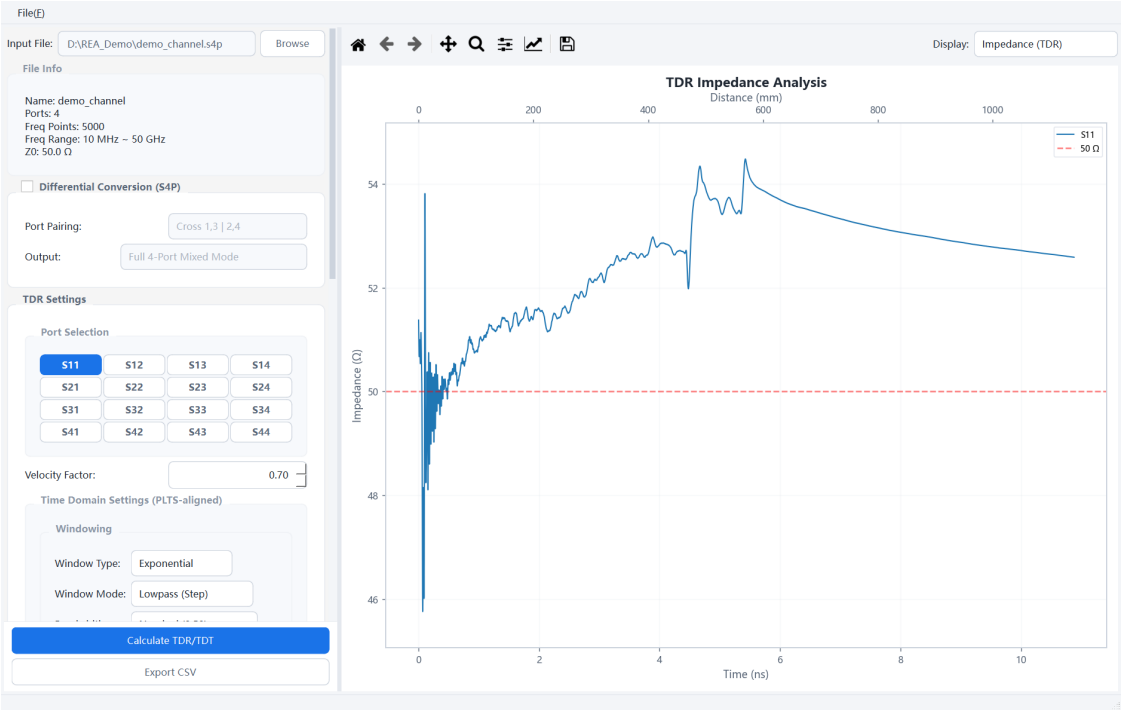

Figure 9-1 TDR — 4-port differential impedance analysis with distance axis

9.1.1 Velocity Factor and Distance Display

The velocity factor (default 0.70) sets the signal propagation speed as a fraction of the speed of light. A secondary X-axis on the TDR plot shows the physical distance (mm) corresponding to the time axis. Adjust the velocity factor to match the dielectric constant of your transmission medium for accurate distance readings.

9.1.2 CSV Export

Click the "Export CSV" button to save the TDR results (time, distance, and impedance columns) as a CSV file for post-processing in external tools such as Excel or MATLAB.

9.2 Smith Chart (PLTS-aligned optimized)

The Smith chart uses a custom Matplotlib component, numerically byte-equivalent to PLTS Smith chart (Re/Im(Γ), Re/Im(Z), VSWR all matched at machine precision). Interactive zoom, pan, and double-click reset are supported.

Chart Type (PLTS chart_type): Z impedance (default) / Y admittance (inverted Smith), switchable via dropdown.

Reference Impedance Z_ref (PLTS Port Reference Impedance): 50Ω SE default, adjustable 1-10000Ω. Quick presets 50/75/100/150Ω. When differential mode enabled, automatically switches to 100Ω (PLTS diff default), and back to 50Ω when disabled.

VSWR Circles (PLTS draw_vswr): toggleable (default ON), shows VSWR=1.5/2/3/5 contours.

Hover Marker: hover over trace to see complete PLTS-style readout — freq, Z=R+jX (Ω), Γ=Re+jIm, |Γ| dB, VSWR, Return Loss.

1. Select S-parameter files (multi-file overlay supported, color-coded).
2. Choose S-parameter to display (S11/S22/S12/S21 + 4-port mixed-mode).
3. Adjust chart type / Z_ref / VSWR display.
4. Chart auto-draws impedance trajectory; hover for detailed values.

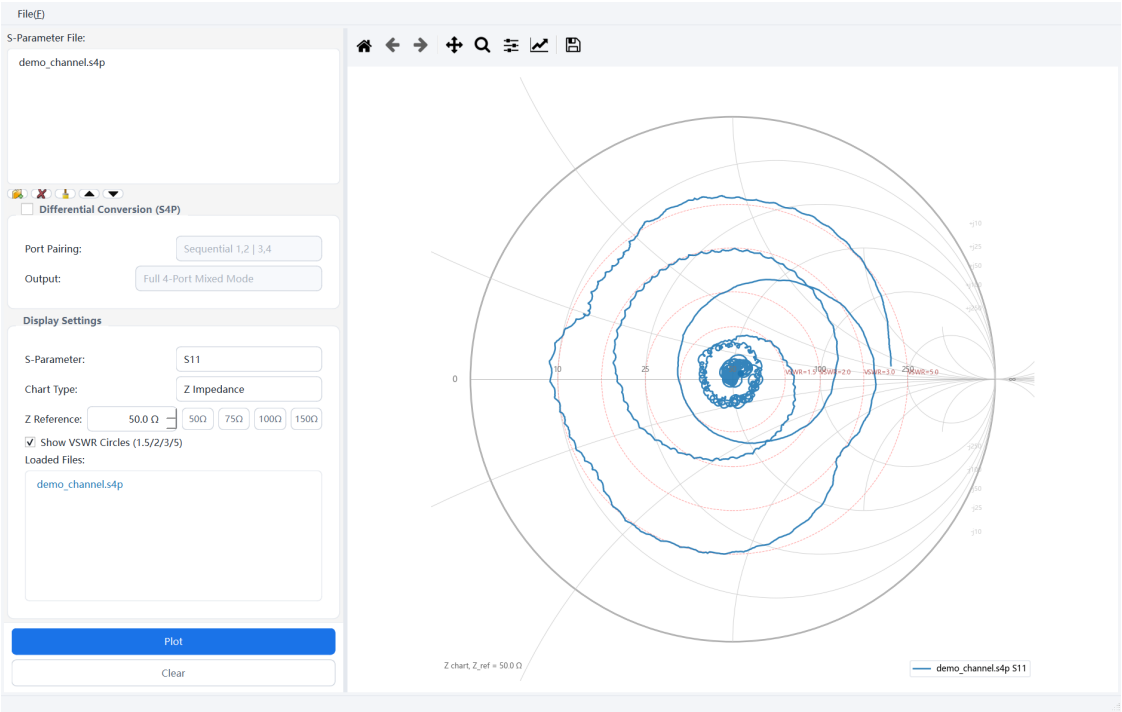

Figure 9-2 Smith Chart — Sdd12 impedance trajectory

9.3 Time-Domain Gating

Time-Domain Gating keeps or removes the response of a chosen time span and transforms back to the frequency domain — used to strip reflections from fixtures, connectors, or other regions you do not care about. Equivalent to PLTS Tools → Time-Domain Gating.

Operation: on the TDR page, enable the "Time-Domain Gating" group in the left panel and set the gate start/stop times.

- **Bandpass** (default): keeps the response inside the gate (= PLTS Bandpass).
- **Bandstop**: removes the response inside the gate (= PLTS Notch).

The corresponding SCPI commands are CONF:TDR:GATE:STATe/TYPE/STARt/STOP (see Chapter 13).

Chapter 10 NRZ/PAM-N Virtual Eye Diagram

The virtual eye diagram is the core feature of this software. It simulates the eye diagram of a high-speed serial signal passing through a channel by convolving a PRBS pattern with the channel S-parameters, evaluating channel signal integrity.

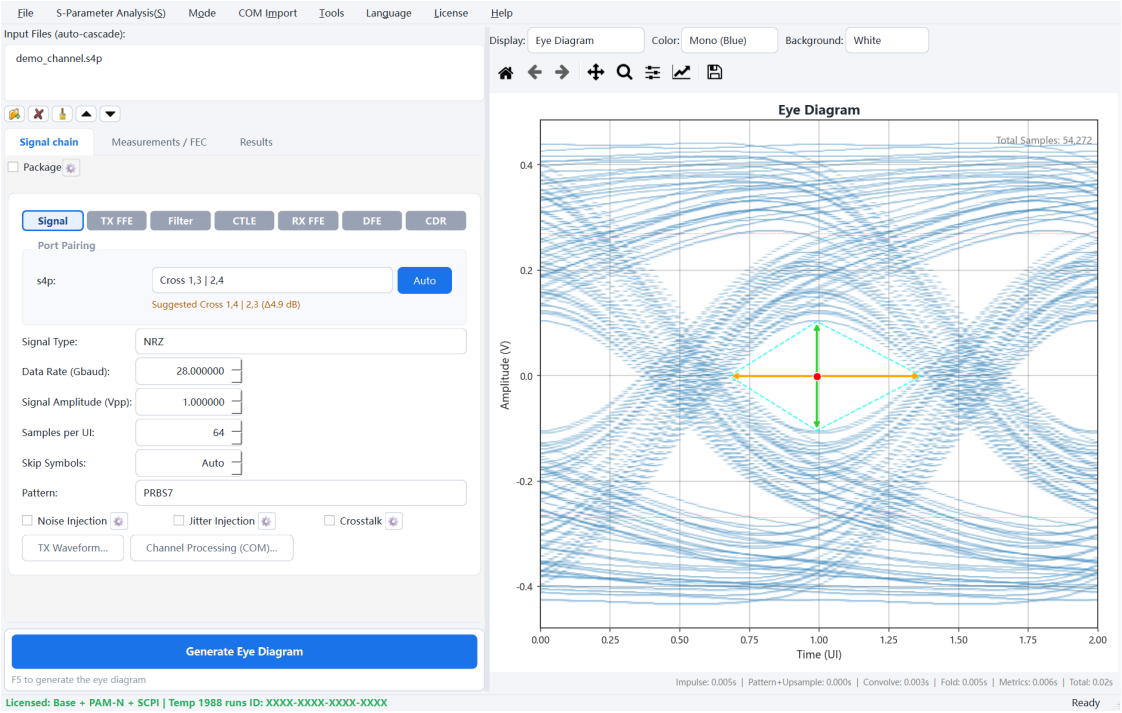

Figure 10-1 Eye analysis main window - Signal Settings page

10.1 Basic Concepts

The left panel has three task pages: "Signal chain", "Measurements / FEC" and "Results". Generate, Pause and Stop remain at the bottom while parameters or long results scroll. Press F5 to generate the eye. Stage buttons open parameter pages; the Enable checkbox inside each page controls whether the stage runs. Set the FEC preset, physical lane count and added BER on the Measurements / FEC page. Leave added BER blank for the preset default.

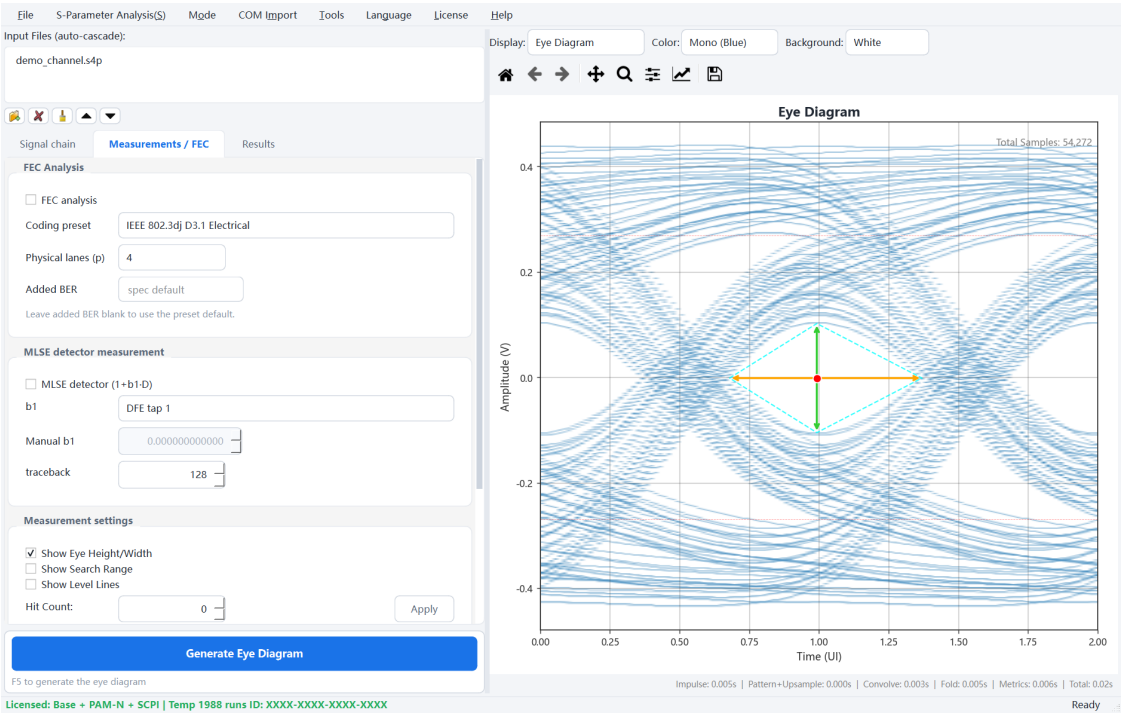

Figure 10-1a Measurements / FEC and the fixed run controls

- **Pattern**: Supports PRBS-7 through PRBS-31, SSPR (OIF-CEI, locked to NRZ), and SSPRQ (IEEE 802.3, locked to PAM4) patterns.
- **Custom Pattern**: Supports loading custom pattern files (.txt) containing space/newline/comma-separated 0/1 bit sequences. Example files (e.g., PRBS9.txt) are provided in the _internal/examples/ directory.
- **Signal Type**: NRZ (2 levels) and PAM-N (N levels, N=3/4/5/6/7/8). PAM-N requires the PAM-N option or a temporary license.
- **Gray Code**: Optional Gray encoding ensures adjacent levels differ by only 1 bit. Effective only for PAM-4/PAM-8 (powers of 2). SSPRQ patterns have built-in Gray encoding unaffected by this toggle.
- **Eye Metrics**: Eye Height, Eye Width — uses 1st/99th percentile statistics to avoid zero-crossing outliers.

10.2 Signal Settings

Configure the following signal parameters in the left control panel:

10.2.1 Basic Parameters

- **Signal Type**: NRZ (2 levels) or PAM-N (N=3/4/5/6/7/8). PAM-N requires the PAM-N option or a temporary license.
- **Data Rate (Gbaud)**: Symbol rate of the serial link, range 0.1–250.0.
- **Pattern**: PRBS order (7–31), SSPR, SSPRQ, or Custom.
- **Samples per UI**: Number of samples per unit interval.

10.2.2 Port Pairing (s2p / s3p / s4p)

The "Port Pairing" group at the top of the eye diagram signal panel presents the appropriate port configuration according to the type of the loaded S-parameter file (a separate TX/RX port row is no longer required):

- **s2p (single-ended 2-port)**: direction 1→2 or 2→1 (mirror S11↔S22). SCPI: CONF:S2P:DIR 0|1
- **s3p (single-ended ↔ differential)**: each s3p file has its own differential pair [(1,2)/(2,3)/(1,3)] and direction (single-ended→differential / differential→single-ended). SCPI: CONF:S3P:PAIR / CONF:S3P:DIR
- **s4p (differential 4-port)**: port pairing scheme (default crossed 13 in / 24 out); the 4 ports are extracted to 2 ports via differential Sdd. SCPI: CONF:DIFF:PAIR. The "**Auto**" button guesses the pairing from low-frequency Sdd21 through-ness minus self-reflection (|Sdd11|, |Sdd22|): the right pairing is ~0 dB through with good return loss; a pairing that splits the two ports of one side leaves only coupling; for two uncoupled symmetric lines the "line B reversed" pairing is algebraically identical to the right one (ties resolve to the crossed convention). The hint turns yellow when it disagrees with the current choice. SCPI: CONF:DIFF:PAIR AUTO / CONF:DIFF:PAIR:AUTO?

10.2.3 Multi-file Mixed Cascade (s2p/s3p/s4p any order)

The "Input Files" list can load multiple S-parameter files and cascade them automatically, supporting any mix of s2p, s3p, and s4p in any order: s4p segments extract differential Sdd by port pairing, s3p segments are reduced to [single-ended, differential] by their differential pair, and s2p segments are cascaded in list order in the 2-port differential domain.

> **Tip:** SCPI multi-file cascade: load the first file with CONF:FILE, then append the rest with CONF:FILE:ADD; RUN and all equalization optimizers (CTLE/FFE/DFE/joint/floating/optical link) use the same chain-building semantics.

10.2.4 Signal Amplitude (Vpp)

The "Amplitude (Vpp)" control sets the peak-to-peak voltage swing of the transmitter output signal. Range: 0.10–5.00 V, default 1.0 V. This value directly scales the eye diagram voltage axis and all voltage-domain metrics (eye height, noise margin).

10.2.5 Rise Time and Fall Time

Open "TX Waveform" on the Signal tab to configure TX edges and Gaussian filtering. "Channel Processing" contains interpolation, causality correction and truncation settings.

Rise time (UI) and Fall time (UI) control the TX edge transition time, range 0.00–0.50 UI, using the 20%-80% definition (the specified time covers the amplitude from 20% to 80%; the actual 0%-100% ramp length = set value / 0.6). Set to 0 for an ideal step; a positive value uses a linear ramp to simulate the edge softening effect caused by limited transmitter bandwidth.

> **Tip:** Rise and fall times can be set independently to model asymmetric TX drivers. Equal rise/fall times are typical for most applications.

10.2.6 Decision Params / COM Noise Budget Sub-windows

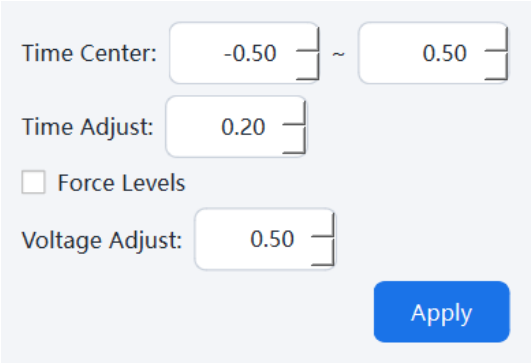

Figure 10-2 Decision parameters sub-window

Below the metrics panel, the "Decision Params" and "COM Noise Budget" buttons each open an independent modeless sub-window (they do not occupy the eye diagram area and can be viewed side by side with the eye):

- **Decision Params**: Time Center (decision-point time search range), Time Adjust, Force Levels, Voltage Adjust; "Apply" re-searches the decision point under the current conditions.
- **COM Noise Budget**: A_DD, DER■, η■, σ_RJ, SNR_TX, σ_N, plus a "Budget→Eye" checkbox; "Apply" recomputes the COM/VEO metrics.

> **Tip:** "Budget→Eye" (default ON) derives and injects the COM statistical noise budget (η■/σ_N/A_DD/SNR_TX/σ_RJ) into the displayed eye so that VEO/SNR_REA reflect the full noise budget and converge toward SNDR_93A; display only, COM_93A unchanged, optimizers isolated. SCPI: CONF:NOIS:BUDGetmap.

Force Levels (within the "Decision Params" sub-window) allows manual specification of the PAM decision threshold levels used for eye metric computation, overriding automatic decision-level detection. When enabled, one spin box per level appears, allowing direct entry of voltage values.

- For NRZ: 2 levels (high and low voltage thresholds).
- For PAM-N: N levels corresponding to the N signal amplitude levels.

> **Note:** Force Levels override the automatically detected levels. Use this only when the automatic level detection produces incorrect results (e.g., severely closed eyes).

- **Channel perturbation / AC common mode (COM 4.x)**: a second parameter group in the same sub-window, named as in the COM 4.x configuration and filled by the COM import; all-zero / κ = 1 means not applied (results unchanged). **Txp/Txn/Rxp/Rxn skew**

(ps): each single-ended port of a 4-port channel gets a phase e^(j2πfτ) before the mixed-mode transform (COM Txpskew etc.) — equal p/n skew is a pure delay, differential skew produces mode conversion and ISI; **κ1/κ2**: scaling of the differential reflections Sdd11/Sdd22 (COM kappa1/kappa2, 0 = reflections at tp0/tp5 omitted), applied before the package cascade; **AC_CM_RMS (mV)** and **ACCM f_max (GHz, 0 = fb)**: TX-side AC common-mode noise source leaking to differential through the 4-port Sdc21 and then the RX chain; the metric σ_ACCM is RSS-combined with σ_N as in COM 4.14 (sigma_ACCM_mV in FETC:EYE:COM?) and, with Budget→Eye on, rendered into the eye with the same convention (band-limited common-mode white noise at N1, one-sided PSD $2A^2/f_max$ flat up to f_max). SCPI: CONF:CHAN:SKEW / CONF:CHAN:KAPPa / CONF:NOIS:ACCM.

• **COM impulse-response truncation threshold / crosstalk ICN (COM 4.x)**: the COM-compatibility row gains a **truncation threshold** (COM configuration Impulse response truncation threshold, default 1e-3): samples of the channel impulse response after the last |h| > peak×thr are zeroed (COM drops them), 0 = no truncation; the COM import fills the configured value and mirrors it into the Expert source node. The crosstalk sub-window gains **ICN amplitude A_ft/A_nt** (COM 4.x, ICN report only, 0 = same as each path amplitude A_fe/A_ne, waveform unchanged). The noise-budget sub-window gains **ICN (CSV, mV)**: the COM import fills ICN_mV from the results CSV; without any aggressor file it enters the COM metric as the Gaussian approximation σ_XT = ICN·σ_x (metrics pane shows XT … (ICN@CSV)) and, with Budget→Eye on, is rendered into the eye with the same convention; once aggressor files are loaded their real crosstalk takes over. The COM import also parses the aggressor base names listed in the CSV file_names column (after the thru; type from fext/next in the name) and pre-fills / registers those found next to the CSV or the victim. SCPI: CONF:CHAN:IRTRunc / CONF:NOIS:ICN / CONF:XTALK:ICN:AMPL.

• **SNR_TXwC0 (COM 4.x)**: the checkbox under SNR_TX in the noise-budget sub-window. When on, σ_TX = (L–1)·A_s/c0/R_LM·10^(–SNR_TX/20), i.e. the transmitter noise is referenced to the un-de-emphasized full swing (with a TX FFE main tap c0 < 1 the noise no longer shrinks with the de-emphasis, more conservative); off = 802.3 equation 93A-30. REA folds it into one effective SNR_TX′ = SNR_TX + 20·log10(c0), so the metric σ_TX and the budget-eye per-symbol amplitude noise follow together; no effect without a TX FFE. Inside a TX FFE optimization c0 is frozen at the current taps (COM uses each candidate's c0); after the optimization everything is recomputed with the new taps. The COM import fills the configured SNR_TXwC0 and mirrors it into the Expert source node. SCPI: CONF:NOIS:SNR_TX:WC0.

• **MLSE reference (COM 4.x)**: with MLSE enabled the COM results CSV reports COM_dB = COM_orig + ΔCOM, where ΔCOM is the Annex 178A analytical detector gain (not a waveform simulation). REA does not model the detector, so its COM_93A corresponds to COM_orig. The import summary and the COM_csv line of the metrics pane show COM_orig and ΔCOM side by side; FETC:EYE:COM? adds CSV_COM_orig_dB,

CSV_delta_COM_dB, CSV_MLSE and REA_minus_CSV_orig_dB next to the existing CSV_COM_dB / REA_minus_CSV_dB. Pass/fail still follows CSV_COM_dB; algorithm agreement is read from REA_minus_CSV_orig_dB.

10.2.7 Level Ratio

The "Level Ratio" checkbox enables non-uniform spacing of PAM signal levels. When enabled, a settings dialog allows specifying the ratio between adjacent level spacings. This models transmitters with intentionally non-uniform level spacing (e.g., PAM4 with RLM optimization).

10.3 Operation Steps

1. Select an S-parameter file (.s2p or .s4p) in the file selector.
2. Configure signal parameters as described in Section 10.2.
3. (Optional) Configure FFE equalizer parameters.
4. Click the "Generate Eye Diagram" button.

   → GUI feedback: The right panel displays the eye diagram histogram and metrics (eye height, eye width, etc.)

To make the effect of each equalizer visible, the screenshots in 10.4-10.9 are all taken on the same lossy channel (about 10 dB at Nyquist, PAM4 53.125 GBaud - the bundled demo/sparams/lossy_channel_10dB.s4p, so you can follow along). The figure below is the unequalized baseline - the eye is nearly closed; each following page enables and optimizes one more equalizer stage on top of the previous one, so the improvement of every stage can be compared step by step.

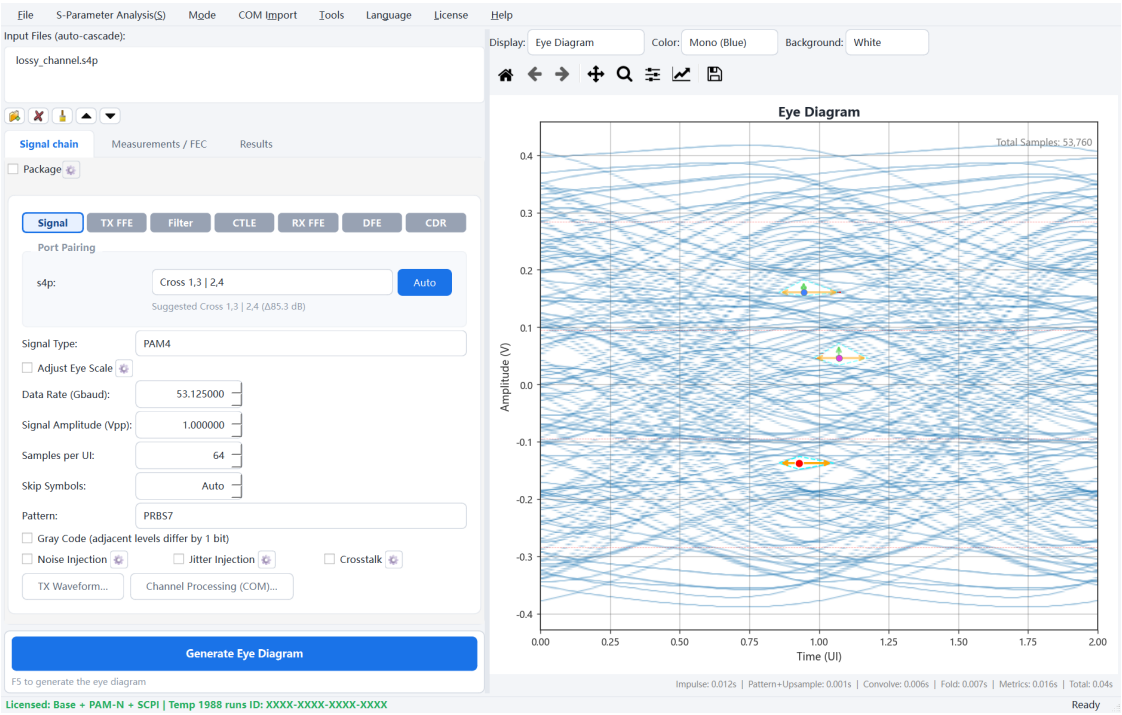

Figure 10-3 Unequalized baseline eye (lossy channel, PAM4) - nearly closed

10.4 FFE Equalizer

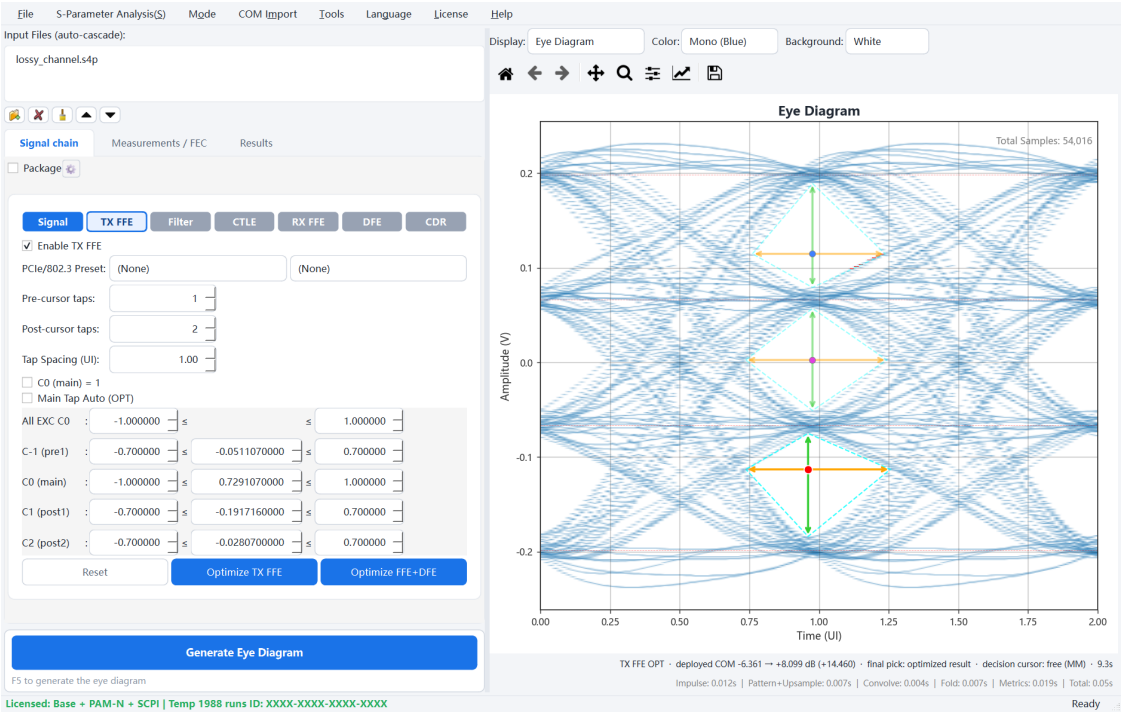

Figure 10-4 TX FFE page - enabled and optimized (compare the Figure 10-3 baseline)

Supports Feed-Forward Equalization (FFE). Tap coefficients can be set manually or optimized automatically.

- **Manual**: Directly enter pre-cursor and post-cursor tap counts and coefficients
- **Optimization bounds**: TX FFE, RX FFE and DFE show coefficients by default. Tick "Show optimization bounds" to expose the limits. Collapsing them does not change the limits or the optimization result.

SCPI DISP:TXFFE:BOUN, DISP:RXFFE:BOUN and DISP:DFE:BOUN with ON / OFF control the same expanded state; append ? to query the current GUI state.

- **Protocol Presets**: Built-in TX FFE presets for PCIe (Gen1-7), USB4 / USB4v2, and 802.3 TX de-emphasis, organized as a two-level "protocol family -> configuration" menu. Selecting a preset auto-fills tap counts, coefficients, and per-tap bounds.
- **Auto Optimization**: The software searches for optimal FFE coefficients targeting COM (link margin). It uses Nelder-Mead by default, switchable to grid search via "File -> Optimization Algorithm" (see 10.14). The search stays strictly within each tap's bounds -- limits set by a preset, the user, or COM Import; with none set the defaults are ±1.0 for the main tap and ±0.7 for the others (main tap = 1; the 802.3 transmitter c(-1) limit corresponds to about 0.74) -- Import are honored by the optimizer.

10.5 RX Filter

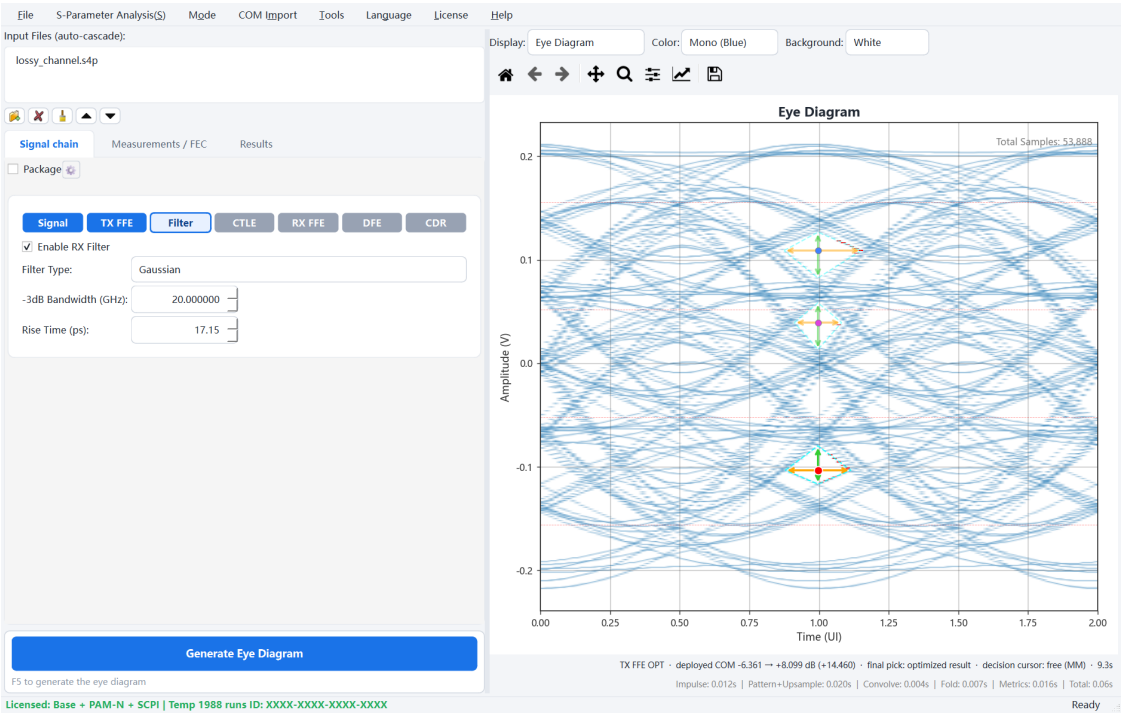

Figure 10-5 RX Filter page - bandwidth-limiting filter enabled

Supports adding a receive-side filter to simulate receiver bandwidth limitations. Filter types include Gaussian, Bessel-Thomson, Butterworth, and RC.

- **Gaussian Filter**: Additionally provides a "Rise Time (ps)" input, linked bidirectionally with the -3dB bandwidth. Mathematical relationship: BW_3dB × T_rise = 0.343 (10%-90% rise time). Changing bandwidth auto-updates rise time, and vice versa.
- **Other Filters**: Only the -3dB bandwidth needs to be set.

10.6 CTLE Equalizer

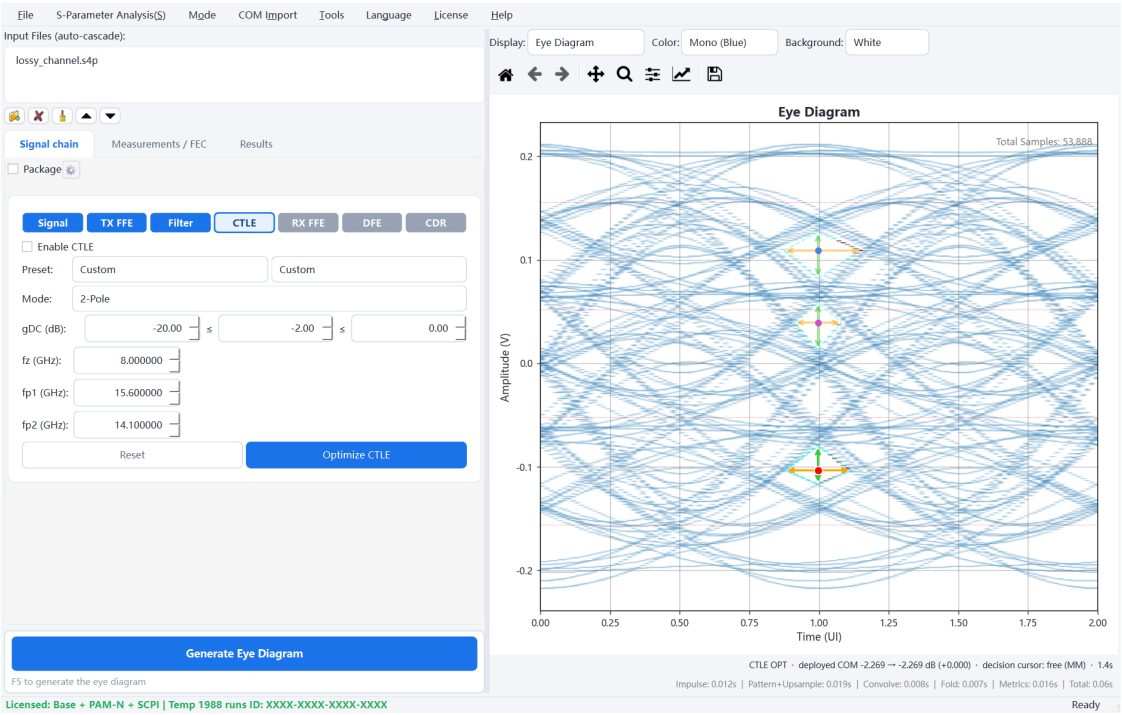

Figure 10-6 CTLE page - enabled and gain-optimized, the eye opens visibly

Supports Continuous Time Linear Equalizer (CTLE), applied after the RX Filter and before the RX FFE. Signal chain: Channel → TX FFE → RX Filter → CTLE → RX FFE.

The CTLE transfer function is based on IEEE 802.3 / OIF-CEI standards, with three modes:

- **2-Pole Mode** (OIF CEI-28G): H(f) = 10^(gDC/20) · (1+jf/fz) / ((1+jf/fp1)(1+jf/fp2)). Parameters: gDC (dB), fz (GHz), fp1 (GHz), fp2 (GHz).
- **3-Pole Extended Mode** (OIF CEI-56G/112G, 802.3dj): Adds low-frequency gain gDC2 and low-frequency pole/zero fLF on top of 2-Pole. Additional parameters: gDC2 (dB), fLF (GHz).
- **3-Pole-2-Zero Mode** (178A): A 3-pole, 2-zero topology with independent control of two gains (g1, g2), two zeros (fz1, fz2), and three poles (fp1, fp2, fp3). Parameters: g1 (dB), g2 (dB), fz1 (GHz), fz2 (GHz), fp1 (GHz), fp2 (GHz), fp3 (GHz).
- **Presets**: Built-in reference CTLE presets for OIF-CEI (28G-VSR / 56G-PAM4 / 112G-PAM4), IEEE 802.3 (802.3dj-200G / 802.3ck), PCIe (Gen1-7), USB3.2 (Gen1/Gen2), and USB4 (Gen2/Gen3), organized as a two-level "protocol family -> configuration" menu. Selecting a preset auto-fills pole/zero frequencies and sets the default value and optimization bounds for gDC (and gDC2 in 3-Pole mode). Selecting an 802.3 family preset (bs/cd CR/KR and C2C, ck CR/KR and C2C, dj 200G) also writes that standard's reference receiver / transmitter tap limits into the DFE, TX FFE (and, for dj, RX FFE) bound controls -- e.g. ck CR/KR DFE tap 1 [0.3, 0.85] and tap 2 [0.05, 0.3], dj DFE [0, 0.85] and TX c(-1) in [-0.34, 0]; existing tap values are clamped into the new bounds. TX FFE bounds follow the standard convention sum|c(k)| = 1 (the coefficients as displayed) and the optimizer searches under that convention; RX FFE bounds follow the COM convention cursor = 1. Users can adjust gDC gain on top of a preset.

- **Reset Button**: Click "Reset" to restore gDC and gDC2 to the current preset's defaults; pole/zero frequency parameters are unaffected.
- **Optimize CTLE Button**: Click "Optimize CTLE" to perform a 2D grid search over gDC and gDC2 (3-Pole mode), finding the gain combination that maximizes COM. The search stays within the current preset's (or the user's) gDC / gDC2 bounds. Other parameters (pole/zero frequencies, FFE, etc.) remain unchanged. The optimal values are auto-filled upon completion. The optimizer finishes with a three-way check on the deployed link (optimized gains / bypass / previous gains) and keeps whichever truly maximizes COM: if bypass wins (e.g. the pole preset does not match the current data rate) CTLE is auto-disabled with a status-bar notice; if the previous gains win they are kept. Under remote control the verdict is reported as SCPI event +301. A bypass verdict is re-checked when the next FFE optimization (TX FFE / RX FFE / joint) completes: the same FFE optimization is repeated with the runner-up CTLE gains and the two end-to-end links are compared on true COM; if "CTLE + re-optimized FFE" now beats the bypassed link, CTLE is re-enabled with that FFE result (both values are shown in the status bar and the +301 event), otherwise bypass is kept. A stand-alone DFE optimization is re-checked the same way: the DFE taps are re-extracted on the CTLE link and the two links are compared on true COM.

10.7 RX FFE Equalizer

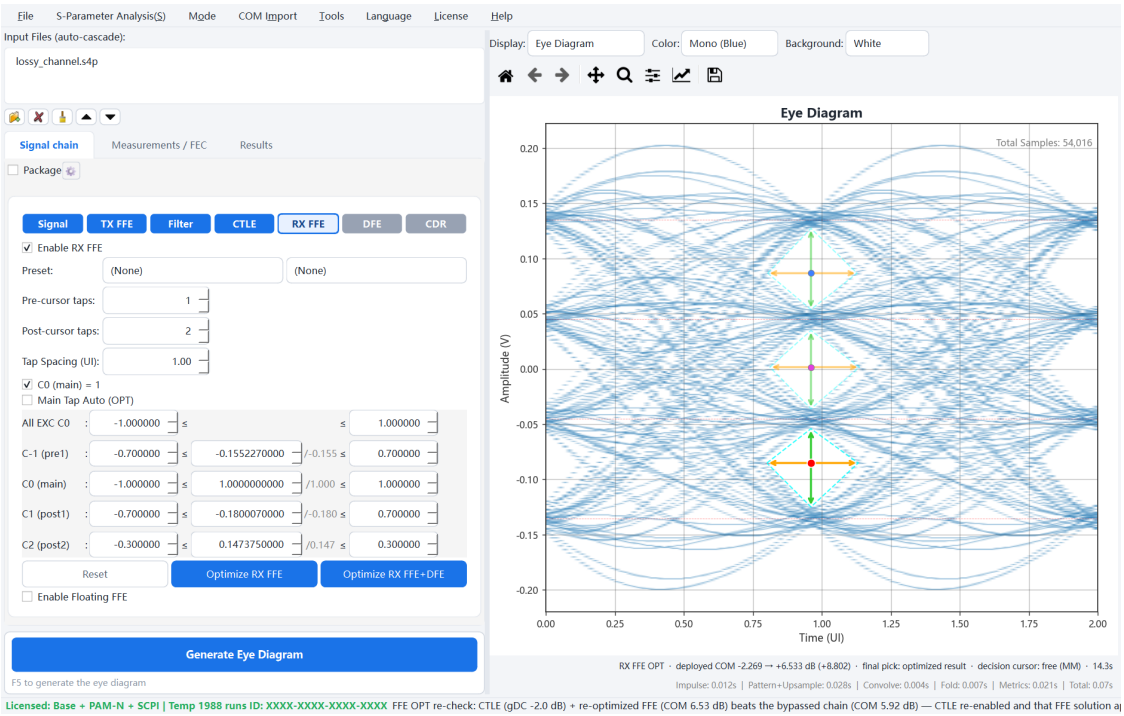

Figure 10-7 RX FFE page - enabled and optimized, eye opens further

Supports receive-side Feed-Forward Equalization (RX FFE), applied after CTLE. Signal chain: Channel → TX FFE → RX Filter → CTLE → RX FFE.

- **Manual**: Directly enter pre-cursor and post-cursor tap counts and coefficients

• **Auto Optimization**: When optimizing RX FFE, TX FFE and CTLE are automatically treated as fixed; and vice versa. The FFE optimizer is CTLE-aware. The search stays within each tap's bounds (receiver defaults when none are set: main tap ±1.0, the two taps adjacent to it ±0.7, all others ±0.3, main tap = 1) -- limits from a preset, the user, or COM Import are strictly honored.

• **Joint FFE+DFE Optimization**: When DFE is enabled, an "Optimize FFE+DFE" button appears on the RX FFE page. Clicking it jointly optimizes RX FFE and DFE: for each candidate FFE tap vector, optimal DFE taps are extracted via the zero-forcing algorithm to maximize combined FFE+DFE eye area. Both RX FFE and DFE tap coefficients are updated upon completion.

• **Optimize DFE**: zero-forcing taps are extracted from the combined pulse response with a joint choice of decision cursor and taps -- every sampling phase within +/-1/2 UI of the pulse peak is tried, scored with the same COM judge as the eye metrics, and the best one wins (on heavily equalized channels the best decision point is usually before the pulse peak); the winning phase also becomes the sampling anchor of the eye / COM.

• **Floating FFE**: Check the "Floating FFE" checkbox to add sparse taps at remote reflection positions, specifically compensating far-end reflection ISI beyond the reach of fixed taps. Configuration parameters (group count N_bg / taps per group N_bf / search range / maximum amplitude bmaxg) are in the Floating FFE settings panel; COM Import auto-populates these from Excel.

Tip: Main switch OFF + Floating ON also supports floating-only mode: When the "Enable RX FFE" main switch is unchecked but "Floating FFE" is checked, only the floating taps are used during eye generation and optimization (no fixed taps applied). This is useful when fixed taps are configured by other means and only far-end reflection compensation is needed.

10.8 DFE Decision Feedback Equalizer

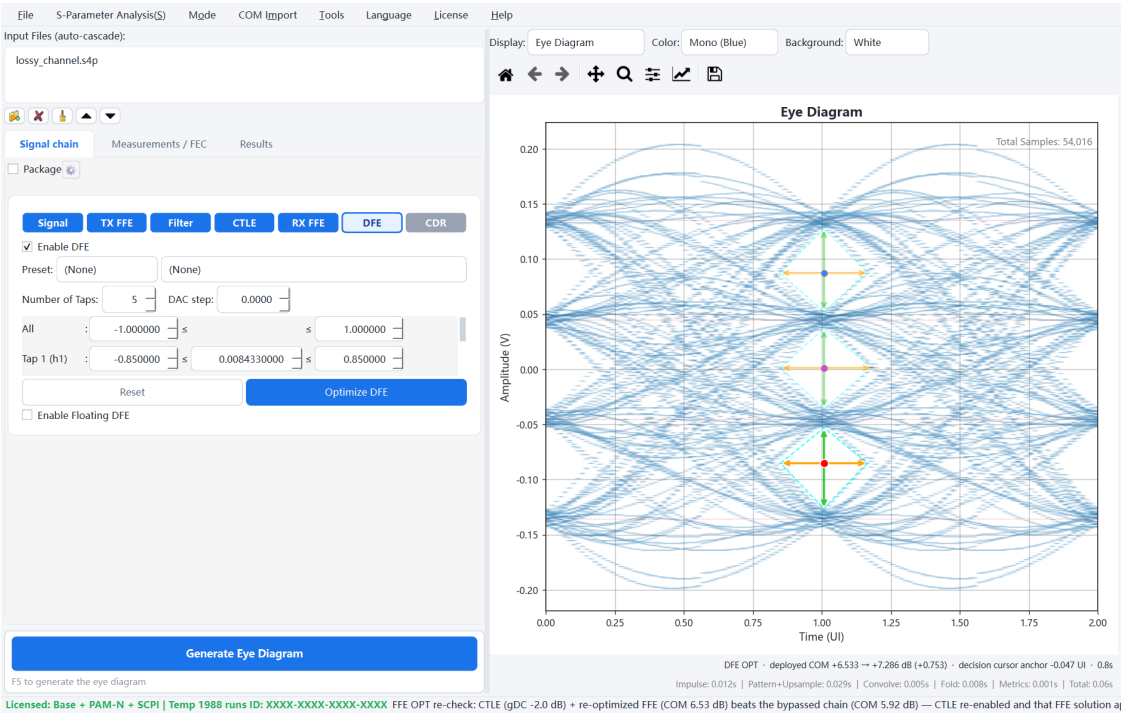

Figure 10-8 DFE page - 5 taps enabled and optimized, residual reflection tails removed

DFE (Decision Feedback Equalizer) is the final equalization stage in the signal chain, performing symbol-by-symbol decision feedback equalization after RX FFE. Full signal chain: Channel → TX FFE → RX Filter → CTLE → RX FFE → DFE.

DFE is fundamentally different from FFE: FFE is a linear feedforward filter that can be pre-convolved into the impulse response; DFE is a nonlinear feedback loop that subtracts the ISI contribution of previous decided symbols at each sample point, then slices to the nearest PAM level.

- **Tap Count**: Supports 1–32 DFE taps via the "Tap Count" control. Each tap corresponds to the ISI cancellation coefficient for a post-cursor symbol.
- **Tap Coefficients**: Each tap ranges from -2.0 to 2.0. Tap 1 (h1) cancels the first post-cursor ISI, Tap 2 (h2) the second, and so on.
- **Combined with FFE/CTLE**: As the final equalizer stage, DFE is typically used in conjunction with CTLE and FFE for optimal eye diagram performance. During FFE/CTLE optimization, DFE is treated as a fixed downstream stage.
- **Reset Button**: Click "Reset" to zero all tap coefficients.
- **Optimize DFE Button**: Click "Optimize DFE" to automatically extract optimal DFE tap coefficients from the channel response. The optimization considers all upstream equalizer settings (TX FFE, RX Filter, CTLE, RX FFE). The extracted taps are clamped to each tap's bounds (set by a preset or COM Import, e.g. the PCIe |d1/h0| ratio cap); the default bounds when none are set are ±0.85 for the first tap with PAM4 (802.3ck / dj reference receiver b_max(1) = 0.85; ±1.0 with NRZ) and ±1.0 for the rest, and joint optimization searches b1 within the same default bound.

- **Floating DFE**: Check the "Floating DFE" checkbox to add sparse DFE taps at remote UI positions, cancelling far-end reflection ISI beyond the reach of fixed DFE taps. Group count, taps per group, search range, and maximum amplitude bmaxg are in the Floating DFE settings panel; COM Import auto-populates these from Excel.

> **Tip: Main DFE OFF + Floating DFE ON also supports floating-only mode**: When the "Enable DFE" main switch is unchecked but "Floating DFE" is checked, only floating DFE taps are used during eye generation and optimization (no fixed DFE taps applied).

10.9 CDR (Clock Data Recovery)

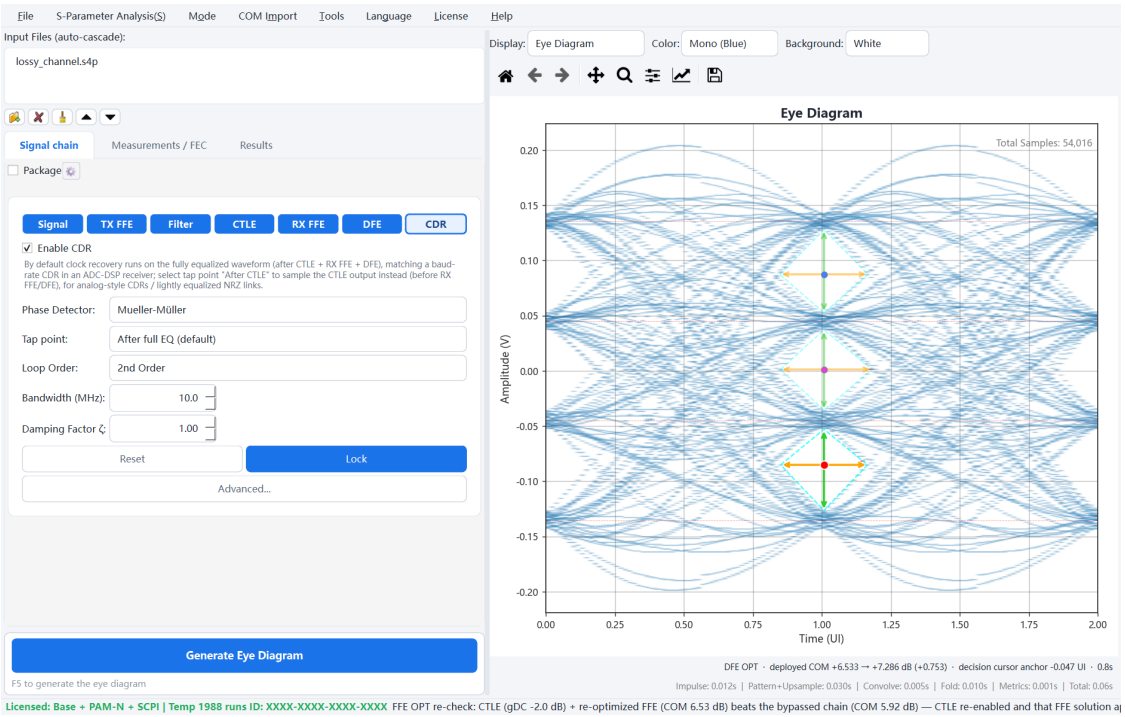

Figure 10-9 CDR page - clock recovery enabled on the fully equalized signal

Two tap points: the default "After full EQ" matches an ADC-DSP baud-rate CDR; "After CTLE" samples the CTLE output instead (before RX FFE/DFE) for analog-style CDRs / lightly equalized NRZ links. On heavily equalized PAM4 the post-CTLE waveform usually carries no timing information - the loop degrades safely to off and reports advisory 305.

CDR simulates a clock data recovery loop that tracks the incoming data stream and adjusts the sampling phase accordingly. When enabled, the eye diagram reflects realistic timing recovery behavior including jitter tracking and SSC tolerance.

10.9.1 Basic CDR Settings

- **Enable/Disable**: Click the "CDR" stage button to open its parameter page, then use the Enable CDR checkbox inside the page to turn CDR on or off.

- **Phase Detector Type**: Choose between "Mueller-Müller" (data-edge correlation, suitable for PAM-N) and "Bang-Bang" (binary phase detector, simpler but with limit-cycle jitter).
- **Loop Order**: "1st Order" (proportional-only control) or "2nd Order" (proportional + integral control, default). 2nd-order loops provide zero steady-state frequency error.
- **Loop Bandwidth (MHz)**: The -3dB bandwidth of the CDR tracking loop, range 0.1–1000.0 MHz, default 10.0 MHz. Higher bandwidth tracks faster jitter but passes more noise.
- **Damping Factor**: Damping ratio of the 2nd-order loop, range 0.10–5.00, default 0.70 (near critically damped). Only visible when Loop Order is 2nd Order.

10.9.2 CDR Lock Button

Click the "Lock" button to enable CDR and immediately generate the eye diagram in one step. This is a convenience shortcut equivalent to checking CDR enable and clicking "Generate Eye Diagram".

Calibration-degenerate notice: the CDR loop self-calibrates its phase detector (MM S-curve) on the **final equalized waveform** (after RX FFE / DFE — the sampling point of a baud-rate CDR in an ADC-DSP receiver). If the equalized eye is still nearly closed and carries no usable timing information, the loop safely disables itself (equivalent to CDR off; jitter is not tracked). The metrics panel then shows a "■ CDR enabled but calibration degenerate" warning line, and under remote control the same fact is reported as SCPI event +305 (both for Normal eye generation and Expert pipeline runs).

10.9.3 CDR Advanced Settings

Click the "Advanced" button to open the CDR Advanced Settings dialog, which provides fine-grained control over the PLL loop parameters:

- **Kp Manual Override**: Check to manually set the proportional gain Kp (range 0.0001–1.0, default 0.001). When unchecked, Kp is automatically computed from the loop bandwidth and damping factor.
- **Ki Manual Override**: Check to manually set the integral gain Ki (range 0.0–0.1, default 0.0001). When unchecked, Ki is automatically computed. Only effective for 2nd-order loops.
- **PD Gain**: Phase detector gain scaling factor, range 0.01–10.0, default 1.0.
- **Phase Clipping (UI)**: Maximum phase error magnitude before clipping, range 0.1–10.0 UI, default 2.0. Limits the phase detector output to prevent large phase jumps.
- **Initial Phase (UI)**: Starting phase offset of the CDR loop, range -2.0–+2.0 UI, default 0.0.
- **Initial Frequency Offset (UI/sym)**: Starting frequency offset, range -0.1–+0.1, default 0.0. Simulates initial clock frequency mismatch.

- **Lock Delay (symbols)**: Number of initial symbols to discard before eye accumulation begins, range 0–10000, default 0. Allows the CDR loop to settle before measurement.

10.9.4 CDR SSC (Spread Spectrum Clocking)

The SSC section within CDR Advanced Settings simulates the TX-side spread spectrum clocking that the CDR must track:

- **Enable**: Check to inject SSC modulation into the CDR test.
- **Frequency (kHz)**: SSC modulation frequency, range 1.0–100.0 kHz, default 33.0 kHz.
- **PPM**: SSC deviation in parts-per-million, range 0–10000 ppm, default 5000 ppm.
- **Mode**: "Down Spread" (frequency modulates downward only) or "Center Spread" (frequency modulates symmetrically around nominal).

10.9.5 CDR SCPI Commands

Command	Description
CONF:CDR:ENAB ON|OFF	Enable/disable CDR
CONF:CDR:TAPP POST|PRE	Tap point (POST = after full EQ, default / PRE = after CTLE)
CONF:CDR:PDTY MM|BB	Phase detector type (Mueller-Müller / Bang-Bang)
CONF:CDR:ORD 1|2	Loop order (1st / 2nd)
CONF:CDR:BAND 10.0	Loop bandwidth (MHz)
CONF:CDR:DAMP 0.70	Damping factor
CONF:CDR:KP 0.001	Manual Kp override (0 = auto)
CONF:CDR:KI 0.0001	Manual Ki override (0 = auto)
CONF:CDR:PDG 1.0	PD gain
CONF:CDR:CLIP 2.0	Phase clipping (UI)
CONF:CDR:PHAS 0.0	Initial phase (UI)
CONF:CDR:FREQ 0.0	Initial frequency offset (UI/sym)
CONF:CDR:DEL 0	Lock delay (symbols)
CONF:SSC:ENAB ON|OFF	CDR SSC enable/disable
CONF:SSC:FREQ 33.0	SSC frequency (kHz)

Command	Description
CONF:SSC:PPM 5000	SSC deviation (ppm)
CONF:SSC:MODE DOWN|CENTER	SSC mode

10.10 Jitter Injection

Jitter injection simulates realistic timing impairments on the transmitted signal. Enable jitter via the "Jitter" checkbox in the signal settings area. When enabled, a settings gear button opens the Jitter Settings dialog.

10.10.1 Random Jitter (RJ)

Random Jitter is specified as an RMS value in UI (unit intervals). Range: 0.000–0.500 UI. RJ is modeled as Gaussian-distributed timing noise added to each symbol transition.

10.10.2 Sinusoidal Jitter (SJ)

Sinusoidal Jitter adds deterministic periodic timing modulation. Up to 6 frequency/amplitude pairs can be configured:

- **Frequency (MHz)**: The SJ modulation frequency, range 0.001–1000.0 MHz.
- **Amplitude (UI)**: Peak-to-peak SJ amplitude, range 0.000–2.000 UI.

Use the "+" button to add SJ pairs (up to 6) and the "–" button to remove the last pair.

10.10.3 Jitter SSC

The SSC section within the Jitter Settings dialog simulates spread-spectrum clocking jitter independently from CDR SSC:

- **Enable**: Check to inject SSC-induced jitter.
- **Frequency (kHz)**: SSC modulation frequency, range 1.0–100.0 kHz, default 33.0 kHz.
- **PPM**: SSC deviation, range 0–10000 ppm, default 5000 ppm.
- **Mode**: "Down Spread" or "Center Spread".

10.10.4 Jitter SCPI Commands

Command	Description
CONF:JITT:RJ 0.010	Set RJ RMS (UI)
CONF:JITT:SJ 100.0:0.1,200.0:0.05	Set SJ pairs (freq_MHz:amp_UI, colon within a pair, comma between pairs)
CONF:JITT:SJ NONE	Clear all SJ pairs

Command	Description
CONF:JITT:SSC:ENAB ON|OFF	Jitter SSC enable/disable
CONF:JITT:SSC:FREQ 33.0	Jitter SSC frequency (kHz)
CONF:JITT:SSC:PPM 5000	Jitter SSC deviation (ppm)
CONF:JITT:SSC:MODE DOWN|CENTER	Jitter SSC mode

10.11 Noise Injection

Noise injection adds Gaussian white noise and/or sinusoidal interference at various points in the signal chain. Enable noise via the "Noise" checkbox. When enabled, a settings gear button opens the Noise Settings dialog.

10.11.1 Gaussian Noise Injection Points

Six injection points (N■ through N■) are available, each with an independent amplitude (Vrms) control. The injection points correspond to positions in the signal chain:

Injection Point	Position in Signal Chain
N■	Before TX FFE (TX noise)
N■	Before Channel (post-TX, pre-channel)
N■	Before RX Filter (channel output)
N■	Before CTLE
N■	Before RX FFE
N■	Before DFE (final noise)

Signal chain with noise injection points: TX +N■ → TX FFE +N■ → Channel +N■ → RX Filter +N■ → CTLE +N■ → RX FFE +N■ → DFE

10.11.2 DM Sinusoidal Noise

Differential-mode (DM) sinusoidal noise adds deterministic interference tones. Up to 6 frequency/amplitude pairs can be configured:

- **Frequency (MHz)**: Interference tone frequency, range 0.001–1000.0 MHz.
- **Amplitude (mVpp)**: Peak-to-peak amplitude of the tone.

10.11.3 CM Sinusoidal Noise

Common-mode (CM) sinusoidal noise simulates interference that appears equally on both lines of a differential pair. Up to 6 frequency/amplitude pairs can be configured, plus a CMRR setting:

- **Frequency and Amplitude**: Same controls as DM noise.
- **CMRR (dB)**: Common-Mode Rejection Ratio, range 0.0–120.0 dB, default 40.0 dB. Determines how much CM noise is converted to DM at the receiver. For differential-mode S4P channels, CMRR is automatically derived from the mixed-mode S-parameters (shown as "Auto" and the CMRR control is disabled).

10.11.4 Noise SCPI Commands

Command	Description
CONF:SIGN:NOIS0–NOIS5 <val>	Set Gaussian noise amplitude at injection point N■–N■ (Vrms)
CONF:NOIS:DM:FREQ0–FREQ5 <MHz>	DM sinusoidal noise frequency (MHz)
CONF:NOIS:DM:AMP0–AMP5 <mVpp>	DM sinusoidal noise amplitude (mVpp)
CONF:NOIS:CM:FREQ0–FREQ5 <MHz>	CM sinusoidal noise frequency (MHz)
CONF:NOIS:CM:AMP0–AMP5 <mVpp>	CM sinusoidal noise amplitude (mVpp)
CONF:NOIS:CM:CMRR 40.0	CMRR (dB)

10.12 Eye Diagram Display Settings

The display toolbar above the eye diagram chart provides the following controls:

- **Display Mode**: Switch between "Eye Diagram" (2D histogram), "Impulse Response", "Pulse Response", "Bathtub" and "Mask Margin" views. The last two are described in 10.12.1; in Mask Margin mode a "Mask…" button appears next to the combo to set the mask shape and size (hidden in the other modes, so the eye / response views keep their layout). The Expert multi-eye "View" combo offers the same five entries per probe.
- **Context menu**: right-click the eye / multi-eye view for "Copy Image" (bitmap to the system clipboard) and "Export Waveform CSV…" (the continuous waveform capture at the decision point, 2048 UI from the fold start by default, two columns time_s,v; not available for streaming long patterns or optical probes). The S-parameter plots have "Copy Image" too. SCPI: CONF:WAVE:SAVE "path" / CONF:PIPE:WAVE:SAVE probe,"path"

- **Color Scheme**: Choose from Monochrome (green/white single-color density map), Heat (warm red-yellow gradient), or Jet (full-spectrum rainbow). Monochrome is the default.
- **Background**: Toggle between White and Black backgrounds. Black background provides better contrast for the eye diagram density map.
- **Stage Overlay Checkboxes**: When viewing pulse/impulse response, floating checkboxes in the chart corner allow toggling visibility of individual signal chain stages (Channel, TX FFE, RX Filter, CTLE, RX FFE, DFE).

10.12.1 Bathtub Curves and Mask Margin

Both views are computed directly from all traces of the current folded eye (the eye algorithm and the other metrics are untouched), matching the S2eye bathtub / eye-mask features.

- **Bathtub (BER vs phase / decision voltage)**: for each eye the traces are split at its decision column into an upper and a lower group by the decision level; sweeping a voltage threshold v gives BER(v) = ½[P(lower > v) + P(upper < v)], sweeping the sampling phase t over the 2 UI window counts per-trace threshold crossings (linearly interpolated between columns) to give BER(t). Beyond the measured part (past the innermost sample of each group) a Gaussian tail is extrapolated: σ is the weighted least-squares slope on the Q scale of the innermost 5 % samples (8–150), anchored at the innermost sample so the curve is continuous with the measured 1/n; solid = measured, dotted = extrapolated. **EH@BER / EW@BER** = width of the contiguous run of the combined curve with BER ≤ target that contains the decision point; the target is the COM noise budget DER_0 (specBER), or 1e-12 when unset, with 1e-6 / 1e-9 / 1e-12 reported too. The trace count n sets the measured floor 1/n and the extrapolation start: longer patterns (PRBS13/15) measure deeper and extrapolate less; a pure-ISI eye without rendered noise still looks Gaussian within the measurable range, so its extrapolation is a conservative (pessimistic) estimate, inherent to trace-based bathtubs (oscilloscope dual-Dirac fits behave the same).

Expert bathtub mode shows one probe at a time. Switch using the probe picker above the plot or the result table; all eyes of the selected probe remain visible. Eyes occupy separate rows in eye-number order: horizontal bathtubs (sampling phase in UI) on the left, vertical bathtubs (decision voltage in V) on the right. Measured and extrapolated segments are drawn separately, with no solid curve underneath extrapolation; dash-dot marks the target BER. Drag a valid eye's BER line vertically with the left mouse button to adjust all eyes continuously; row titles show the current EH or EW and the overall opening remains the worst eye. Release restores adaptive bounds. Toolbar pan/zoom takes precedence over BER dragging. Normal, Expert and probe details remain linked. Exact entry uses specBER / CONF:NOIS:BER; FETC:EYE:BATH? and FETC:PIPE:BATH? report the same target. Dragging reads openings from existing curves without recomputing waveforms or changing the measured/extrapolated convention. Horizontal ranges retain transition shoulders with padding and a common phase window across eyes; the BER range follows the target with one decade below it. Autoscale and Home restore these bounds. If any expected eye lacks samples, usable per-eye curves remain

visible with the usable eye count and an insufficient-data message. Overall EH@BER / EW@BER are unavailable, rather than reported as zero. The SCPI summary returns available=0,status=INSUFFICIENT_DATA; querying before generating an eye retains the empty response and No data error. Usable per-eye curves remain queryable when the overall result is incomplete. Eye indices start at 1; a missing or out-of-range eye returns an empty response without selecting another eye.

- **Mask margin (eye + mask)**: the "Mask…" button sets the shape (rectangle / diamond / hexagon), mask height EH (mV), mask width EW (UI; the rectangle width, the top/bottom edges of the hexagon) and the reference-line width EW@vref (UI; diamond and hexagon width at the decision level). The mask is centred on each eye's decision point (phase, level); any sample inside the polygon (hits > 0, transition traces included) is a FAIL. Margins: top/bottom = minimum voltage distance from the mask top/bottom edge to the inner eye contour, left/right = phase distance from the mask left/right vertex to the nearest inner crossing (hexagon: the smaller of the EH-edge line and the reference line); positive means the eye encloses the mask. The eye height inside the mask width and the eye width at the decision level (rectangle/hexagon: at the two vref ± EH/2 lines) are reported too. Mask parameters persist with the session and are shared with the Expert multi-eye view.

SCPI	Description
CONF:MASK:TYPE none|rect|diamond|hex	Mask shape (none = off, default); :TYPE? queries
CONF:MASK:EH <V> / :EW <UI> / :EWVref <UI>	Mask height (V) / width (UI) / reference-line width (UI), each with a query form
FETC:EYE:BATH?	Bathtub summary: available, target_ber, n_traces, eh_at_ber_v, ew_at_ber_ui, eh_1e-6/9/12_v, ew_1e-6/9/12_ui, eyeN_sigma_v_lo/hi, eyeN_sigma_t_l/r_ui, eyeN_floor
FETC:EYE:BATH:CURV? V|T[,eye]	Curve data x:ber;x:ber;… (V: volts; T: phase in UI; eye index from 1)
FETC:EYE:MASK?	Mask margin: type, eh_v, ew_ui, ew_vref_ui, pass, hits, mask_eh_v, mask_ew_ui, worst_margin_v/ui, eyeN_pass/hits/margin_left/right_ui/margin_top/bottom_v
FETC:PIPE:BATH? / :BATH:CURV? / :MASK? <probe>	Same queries for an Expert probe (probe id first)
DEBU:GUI:EYE:MODE 0..4 / DEBU:GUI:MEYE:MODE 0..4	Drive the Normal display combo / multi-eye view combo (0 eye, 1 impulse, 2 pulse, 3 bathtub, 4 mask)

10.13 Optimization Sampling Density

Via menu bar "File" -> "Optimization Algorithm" -> "Optimization Sampling Density", choose 16 pts/UI (Fast), 32 pts/UI (Default), or 64 pts/UI (Precise). This controls the per-UI sampling used internally while optimizing FFE/CTLE and other equalizers.

> **Tip:** 64 pts/UI suits many-tap FFE optimization (>= 8 taps) and takes about 4x the time of 16 pts/UI. The default 32 pts/UI is recommended for daily use. The setting is saved automatically across restarts.

10.14 Optimization Algorithm Selection

Via menu bar "File" → "Optimization Algorithm", choose the search algorithm and COM-related parameters for equalizer optimization. The optimization objective is fixed to COM (link-margin metric, same source as the COM_93A shown in the results panel).

Menu Item	Description	Default
Nelder-Mead (Fast)	Fast-converging automatic search; fast and stable	Default
Grid Search	Exhaustive tap sweep by step (submenus: step 0.005~0.05 or custom; Local Search = refinement span around the best point; Candidate cap = max valid candidates evaluated per stage, the excess trimmed nearest-to-seed first — a count, not a time limit, so machines of different speed reach the same result)	Step 0.02, LS 35, cap 1M
COM Top-N Cursors	PDF top-N cursors for fast COM evaluation; larger N is more accurate but slightly slower	32

> **Tip:** Both algorithms automatically use local hardware resources (GPU memory / CPU cores) for acceleration. Nelder-Mead is recommended for daily use; Grid Search suits exhaustive-verification scenarios. All optimizations target COM; the pre-optimization parameters (and the COM-import baseline) enter the final selection together with the optimized result, judged by the same metric on the deployed response. The summary reports the actual before/after values and explains no improvement or failure (10.14.2). After changing tap structure, stage switches or search bounds, regenerate the eye to check the result. Fixed and floating taps are optimized separately, preserving the other segment.

10.14.1 Optimization Bounds

Every adjustable parameter of the CTLE (gDC / gDC2), TX FFE, RX FFE, and DFE carries lower/upper bounds. Selecting a protocol preset (PCIe / USB / 802.3 / OIF-CEI) or importing a COM configuration writes these bounds into the corresponding controls; they can also be edited manually in each equalizer panel.

Auto-optimization of all four equalizers -- whether triggered from a GUI button or a SCPI command -- searches strictly within the bounds and never exceeds them. For example, a PCIe DFE preset writes the tap-1 |d1/h0| ratio cap (PCIe5/6/7 = 0.8 / 0.55 / 0.5) into the DFE tap-1 bound, and the DFE optimization result will not exceed it.

> **Tip:** The CTLE gDC / gDC2 optimization bounds can be set individually via the SCPI commands CONF:CTLE:GAIN:MIN|MAX and CONF:CTLE:GAIN2:MIN|MAX; the per-tap bounds of FFE / RX FFE / DFE are configured automatically by a preset or COM Import.

10.14.2 Optimization Summary and Events

After every optimization (CTLE / TX FFE / TX FFE+DFE joint / RX FFE / RX FFE+DFE joint / DFE) finishes and the eye is regenerated, an **optimization summary** line appears to the right of the eye, above the timing line: kind - deployed COM before -> after (delta) - final pick (optimized result / start / baseline pair; joint runs also show the number of starts) - decision cursor (anchor offset or free MM) - elapsed time - event count (hover to read them). It follows the UI language and is cleared by *RST.

Optimization is never silent: when the final selection keeps the start or the baseline pair, when the result merely ties the start (taps not improved), when the DFE optimizer has to clip its zero-forcing taps to the tap bounds (for example the 802.3ck-C2C preset gives the first DFE tap a standard lower bound of 0.3 -- use the joint RX FFE+DFE optimization then), when the deployed COM after an optimization is below the value before, or when the optimizer degrades internally (GPU batch failure falling back to CPU, a failed bypass re-check keeping the bypass solution), an **event** is recorded. Events show in the status bar (last 3), in the summary event count (hover), and in the SCPI error queue as positive code +302 (popped by SYST:ERR?, alongside the +301 verdict advisories). Event text follows the UI language. A failed optimization renders the summary in red (" failed: reason") and reports -300 over SCPI.

> **Tip:** The former "File -> Optimization Algorithm -> Allow COM Degradation" menu and the CONF:OPT:DEGRade command were removed. The software compares the optimized candidates, start and imported baseline with one judge, and reports the actual deployed result in the summary. A running optimization can be stopped from the panel button or by SCPI ABORt (including SCPI-triggered FFE / RX FFE / CTLE optimizations); parameters stay unchanged after a stop.

10.15 GPU Acceleration

GPU / CUDA acceleration is free and enabled by default for all users, with no option license required. After installing CuPy on CUDA-capable hardware, it can be toggled via menu bar "Tools" -> "Enable CUDA Acceleration"; "Tools" -> "CUDA Acceleration Status" shows detection details. GPU acceleration significantly speeds up eye diagram computation for high-order PRBS.

> **Tip:** GPU and CPU computation paths produce identical eye diagram metrics. Toggling GPU acceleration does not affect results, only speed.

10.16 Streaming Processing

High-order PRBS (>15) automatically uses streaming chunk processing: completing the eye statistics of very long patterns in bounded memory, with results identical to whole-signal computation. Streaming mode runs asynchronously via QThread with a progress bar.

10.17 Crosstalk Injection

Crosstalk injection adds FEXT (far-end) and NEXT (near-end) crosstalk paths to simulate multi-channel interference on eye diagrams.

10.17.1 Basic Operation

- Click "Crosstalk" button in the eye diagram tab to open the injection dialog
- Click Add FEXT / Add NEXT to load crosstalk S-parameter files
- S4P files support independent differential port pairing (TX+/TX-/RX+/RX-)
- Each crosstalk path has configurable voltage (Vpp) and delay

10.17.2 ICN Calculation

ICN (Integrated Crosstalk Noise) follows the IEEE 802.3 COM standard:

- Crosstalk signals pass through victim RX equalization (RX Filter → CTLE → RX FFE)
- ICN includes sinc^2 sample-and-hold power weighting
- FEXT ICN additionally includes TX FFE power weighting (consistent with the COM reference implementation)
- NEXT ICN does NOT include TX FFE (per COM standard)
- ICN is scaled by aggressor voltage: $ICN_actual = A_x \times ICN_normalized$

10.17.3 COM Noise Budget Impact

When crosstalk is enabled, ICN contributes to the COM 93A noise budget via $\sigma_XT = ICN \times \sqrt{(L^2-1)/(3(L-1)^2)}$, where L is the number of PAM levels. $\sigma_total = RSS(\sigma_ISI, \sigma_J, \sigma_XT,$

σ_N, σ_TX). The eye metrics panel shows ICN components (FEXT/NEXT listed separately), and the COM breakdown line includes the XT component.

10.17.4 Eye Three-Metric Layering (COM_93A / SNDR_93A / SNR_REA)

REA provides three semantically-independent dB metrics, each corresponding to a distinct "signal vs noise" dimension. Each metric has an explicit 20·log10(signal/noise) formula:

- **COM_93A** = 20·log10(As/A_ni), **BER-conditional**. Based on IEEE 802.3 93A PDF integration + CDF lookup for A_ni @ target BER. SCPI field COM_93A=. This is the protocol-standard COM, aligned with COM Tool / .m.
- **SNDR_93A** = 20·log10(As/σ_total), **BER-invariant**. Based on COM 93A noise budget σ_total = RSS(σ_ISI, σ_J, σ_TX, σ_N, σ_XT), including ISI distortion + full noise budget. Matches IEEE 802.3ck Annex 162B SNDR. SCPI field SNDR_93A_dB=. Invariant to target_BER (decoupled from BER).
- **SNR_REA** = 20·log10(As/σ_eye), **BER-invariant**. The equivalent noise sigma_eye is extracted from the decision-point statistics of the rendered eye. SCPI fields SNR_REA_dB= + sigma_eye_mV=. This is the actual eye-diagram amplitude SNR, complementing the protocol σ_total: σ_eye is typically < σ_total (eye statistics do not include full budget, only ISI residual + injected sim noise).
- **VEO_REA**, **BER-conditional**, in mV: the vertical eye opening extrapolated from eye statistics to the target BER. SCPI fields VEO_mV= + VEO_PER_EYE=.

Typical ordering: SNR_REA > SNDR_93A > COM_93A (gap 5–22 dB depending on scenario). Engineers can use per-σ SNR sub-components (SNR_X = 20·log10(As/σ_X)) to identify the dominant noise: if SNR_ISI is close to SNDR_93A, ISI is the bottleneck; if SNR_N is lowest, RX thermal noise dominates (typical for high-loss channels).

Note: The legacy field COM_REA has been removed (semantic overlap with COM_93A). Migrate user scripts to SNR_REA_dB= or COM_93A=.

10.17.5 Channel Quality Lines (the "Channel @Nyq" lines of the pane)

After an eye is generated the metrics pane appends, after ICN, a report card of the channel itself (same renderer for Normal and Expert probes; SCPI FETC:EYE:CHAN? / FETC:PIPE:CHAN? share the source). All dB values use the IEEE 802.3 **positive-loss** convention (bigger is better; opposite sign to 20·log10|S| on the S-parameter plots). They are taken from the passive channel before the eye computation (package cascade included, terminations / RX filter excluded); Nyquist = symbol rate / 2; crosstalk aggressors use the same package treatment as the victim.

Metric	Meaning / source
IL @Nyq	–20·log10|Sdd21| at Nyquist; when Nyquist exceeds the file bandwidth the endpoint is used and flagged

Metric	Meaning / source
fit / ILD max / FOM_ILD	IEEE 802.3-2022 93A.3 fitted insertion loss a0+a1√f+a2f+a4f² (weights 10^(–IL/20), band 50 MHz–fb); 93A.4 ILD = IL – IL_fitted, maximum absolute value in band; FOM_ILD = √(1/N Σ W·ILD²), W = sinc²(f/fb)/(1+(f/ft)■)/(1+(f/fr)■), ft = 0.2365/Tt (TX Gaussian Tr), fr = RX filter bandwidth
RL @Nyq TX/RX, worst in band	–20·log10|Sdd11| / |Sdd22|; worst in band = minimum of both ports
delay	Least-squares slope of the unwrapped Sdd21 phase versus ω (50 MHz–Nyquist)
Crosstalk PSXT / FEXT / NEXT / ICR	Power-sum crosstalk loss –10·log10 Σ|S_xt|² over the loaded FEXT/NEXT aggressors; ICR = PSXT – IL (insertion-loss-to-crosstalk ratio, same form as 802.3 92.11 / 93.9)
SDC21 / SCD21	Mode-conversion loss of the 4-port mixed-mode network (differential→common / common→differential)
intra-pair skew	Difference of the phase delays of the two single-ended through paths (p1+→p2+, p1–→p2–)
h-1 / h0 / h+1, ISI/h0	Cursor voltages of a full-swing 1-UI pulse after equalisation (stage before the DFE) and the normalised ISI Σ|h_k≠0|/|h0|; "post-DFE" is the residual after subtracting the DFE taps

Tip: These map one-to-one onto the S2eye runstats columns (ILS/ILS_fit/ILD/RL/XTK/ICR/DLY/SCD21/SDC21/hm1/h0/hp1/norm_isi), except that REA counts positive loss. With a COM import they can be cross-checked against the COM tool's channel FOMs.

10.18 COM Import

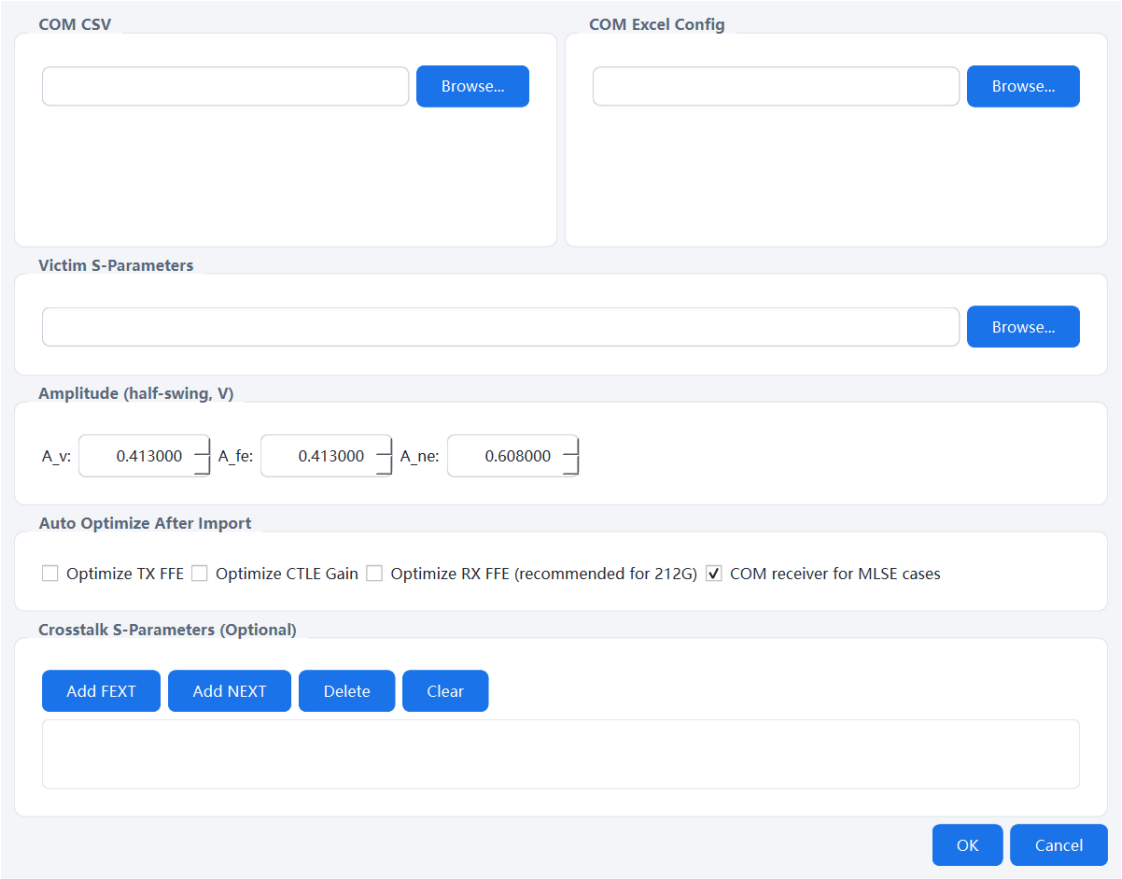

Figure 10-10 COM Import dialog

COM Import loads all parameters from IEEE 802.3 COM tool output CSV and configuration Excel, including signal settings, CTLE/FFE/DFE configuration, Package Model (TX/RX), PCB traces, TX Gaussian filter, RX filter, noise budget parameters, and port pairing.

- CSV file: COM optimization results (CTLE gains, FFE/DFE taps, frequencies)
- Excel file: A_v, RX Filter, TX Gaussian, Package Model configuration
- CTLE mode auto-detection: selects 2-Pole / 3-Pole / 3P2Z based on fz2/fp3
- Sampling phase: PEAK (COM standard) or PULSE (optimized)

Steps:

1. Click "COM Import" in the menu bar to open the dialog (Figure 10-10).
2. Pick the results CSV in the "COM CSV" field - channel/rate/COM preview appears below.
3. Pick the configuration Excel in "COM Excel Config".
4. Pick the victim channel .s4p.
5. (Optional) tick auto-optimize, then OK - all parameters are imported and a baseline eye is generated automatically.

> **Tip:** Bundled templates: demo/com_import/ ships three complete sets (50G / 120G / 212G) of CSV + Excel + channel - try the steps above directly (see 3.4).

10.18.1 Package Model

Package model cascades in the 4-port domain before Sdd extraction. Supports separate TX/RX package parameters (gamma, tau, Z_c), PCB board traces (C_0/C_1 capacitors, Z_c impedance), and bump/via/die pad parasitics.

Chapter 11 Expert Mode (Pipeline Designer)

Normal mode uses a fixed signal chain: source, channel, equalizers, eye. Expert mode opens that chain up into a free node canvas - arrange signal-processing blocks in any order, build parallel branches, and compare equalization schemes side by side with multiple eye probes, just like on a real bench.

Version 2.0.28 configuration save/load preserves floating FFE/DFE coefficients, positions and decision timing. Loading an older configuration without these fields clears the previous floating layout. The distribution includes demo/floating_ffe/: channels contains S-parameters, configs contains reproducible configurations, images contains reference figures, and README.txt maps the files. Keep the whole directory structure when moving it; channel paths are relative to each config file. Regenerate the eye after loading and compare the results. Re-optimization may select a different layout.

11.1 Entering and Leaving

Menu bar "Mode" -> "Expert" to enter; "Normal" to return. The two modes are fully independent and never affect each other: switching simply moves between the two workspaces - the Expert canvas is saved on exit and restored on re-entry, and Expert edits never write back to the Normal configuration. A config file saved via "File -> Save Config" (Ctrl+S) carries both states: the Normal panel parameters, the Expert canvas (nodes/connections/parameters/positions) and the active mode - loading it restores the mode and the canvas together.

- **Copy Normal Config to Expert** ("Mode" menu button, visible in Normal mode only): a one-shot copy that converts the current Normal configuration (channel files, package, enabled equalizers, noise settings) into an equivalent pipeline in physical signal-chain order, overwriting the Expert canvas - after copying, enter Expert to see an eye equivalent to Normal with no reconfiguration. This button is the only bridge between the two modes.

> **Note:** The Expert canvas does not follow later Normal changes automatically - after adjusting the Normal configuration, click "Copy Normal Config to Expert" again to bring it over (this overwrites the current Expert canvas).

11.2 Canvas Operations

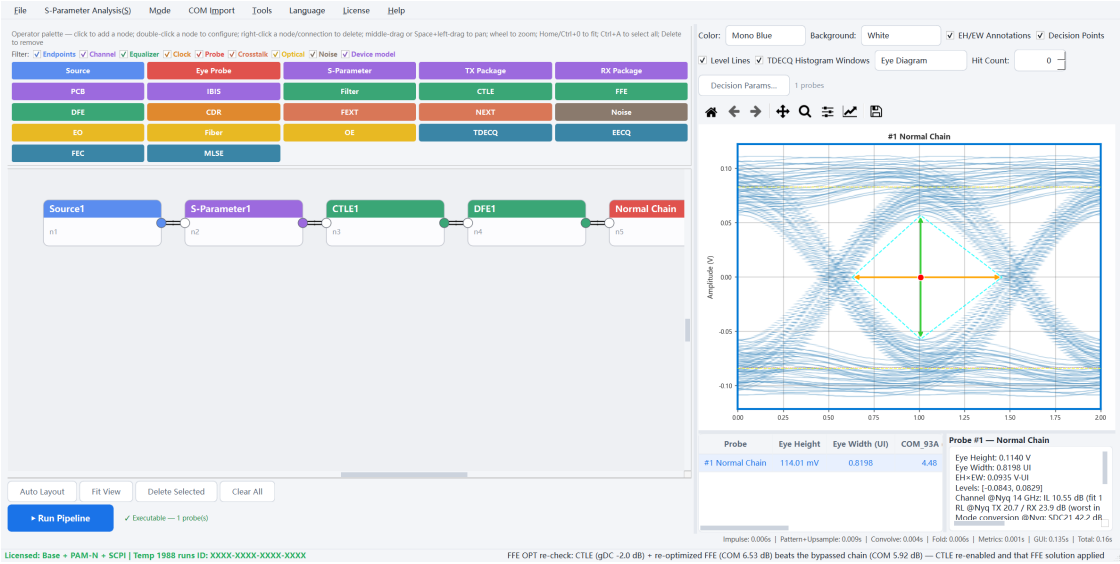

Figure 11-1 Expert mode - node canvas and multi-probe eyes

Action	Gesture
Add a node	Click a type button in the palette above the canvas
Connect	Drag from an output dot to a downstream input dot
Edit parameters	Double-click the node to open its dialog
Node/edge menu	Right-click a node or edge (delete, optimize...)
Pan	Middle-drag, or hold Space + left-drag
Zoom	Mouse wheel; Home or Ctrl+0 fits the view (always 1:1 — nodes never grow with the window, a larger window simply shows more canvas)
Select / delete	Ctrl+A selects all; Delete removes selection
Tidy layout	"Auto Layout" button (topological, left to right)

11.3 Node Types

Node	Purpose	Per chain
Source	Signal source (type/rate/pattern/amplitude/jitter/TX Gaussian)	1 global
S-Parameter	Channel / cable / fixture S-parameter segment	multiple
TX/RX Package, PCB	Package and board-trace models	1 each

Node	Purpose	Per chain
FFE	Feed-forward equalizer; side parameter picks TX or RX	1 TX + 1 RX
Filter	RX bandwidth-limiting filter (Bessel/Butterworth...)	at most 1
CTLE	Continuous-time linear equalizer (2P/3P/3P2Z/PCIe)	at most 1
DFE	Decision feedback equalizer	at most 1
CDR	Clock data recovery	at most 1
IBIS	IBIS / IBIS-AMI device model (IBIS 8.0: .ibs linearised or Ts4file analog front end + AMI executable, see 11.6)	TX/RX alternating, repeater pairs allowed
FEXT / NEXT	Far-/near-end crosstalk injectors	multiple
Noise	White/sinusoidal noise injection at any position	multiple
EO / Fiber / OE	Optical link blocks (see Chapter 12)	1 each per optical chain
TDECQ	802.3 reference-receiver metric (see Chapter 12)	terminal node
Eye Probe	Probe - one eye diagram per probe	1 per chain end

Topology rules: exactly one Source per pipeline; the signal may fan out into parallel branches (e.g. compare "CTLE only" vs "CTLE+DFE" behind the same channel); every branch ends in a probe (or TDECQ).

11.4 Run and Results

The multi-eye toolbar keeps the view selector, "Plot Settings..." and probe count. The Appearance tab of the settings window contains color, background, EH/EW annotations, decision points, level lines and TDECQ histogram-window switches. Decision Settings contains hit count, search conditions, forced levels and manual decision points. Appearance changes update immediately; decision controls retain their existing apply behavior. Closing the window or pressing Esc preserves its values. Normal parameter edits synchronize with SCPI queries, including floating FFE/DFE, level ratios, forced levels and RX rise time. Setting CDR KP/KI enables the matching override; OFF clears its checkbox. The frequency-offset query retains four decimals. Package, measurement-search, DFE-step and automatic-main-tap edits also

update queries. Apply recalculates existing results; editing one parameter preserves the precision of other imported values.

Click "Run Pipeline" to execute. Every probe renders one eye in the multi-eye area; the table below lists eye height / eye width / COM_93A / SNDR_93A / VEO / SNR_REA per probe - click a row for the full details of that probe.

For long runs (optical chains, long patterns) a **progress bar** appears below the canvas: percentage, elapsed and estimated remaining time update live, and the status bar shows the node currently executing. The "■ Pause" and "✕ Cancel" buttons next to it freeze or stop the run at any time - cancelling does not consume a trial credit, and the eyes keep the previous run's results. A remote-controlled CONF:PIPE:RUN lights up the same progress display.

> **Note:** Any parameter or wiring change marks the displayed eyes as stale (yellow "re-run required" banner). Click "Run Pipeline" again to refresh.

11.5 Per-node Optimization

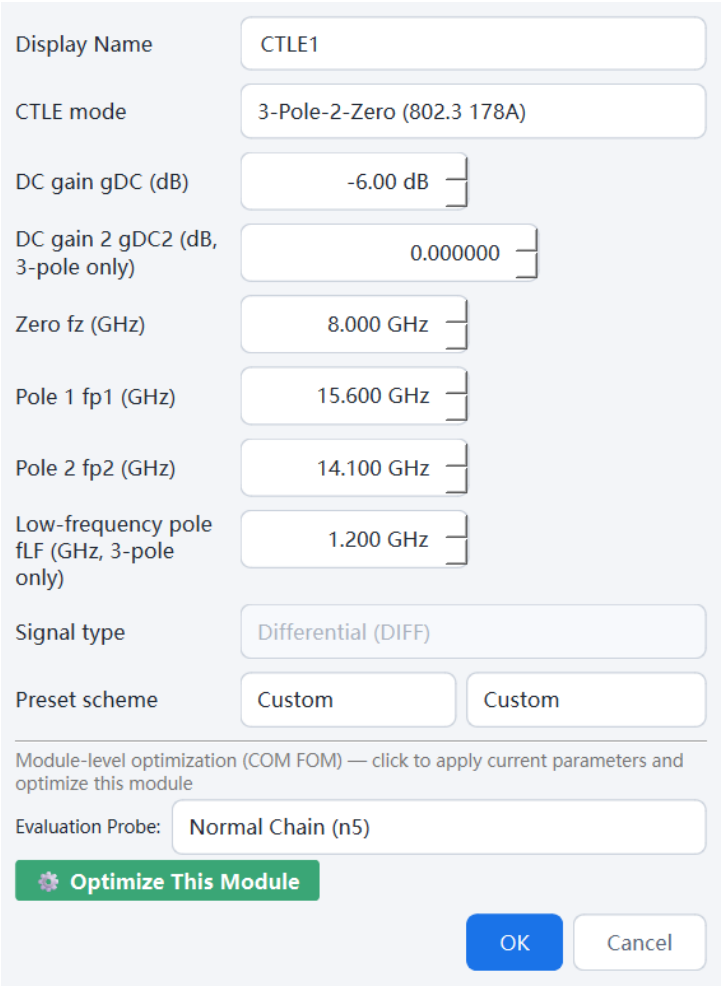

Figure 11-2 Node dialog (CTLE example with optimize buttons)

Double-click an FFE / CTLE / DFE node to open its dialog, which offers "NM Optimize" and "Grid Optimize" for FFE, or "Optimize This Module" for CTLE / DFE (or right-click the node ->

"Optimize this module"). The optimizer evaluates the complete chain down to the probe below that node with COM as the target - results match Normal-mode optimization. New parameters are written back to the node automatically. CTLE optimization also runs the three-way check (optimized gains / bypass / previous gains): previous gains are restored automatically when they are truly better; when bypass wins the canvas topology is left untouched and a status-bar notice suggests removing the CTLE node or checking the pole preset. The three-way check uses the same noise budget as the probe display, and once an FFE on that chain is optimized the "keep CTLE" vs "remove CTLE" links are compared again on true COM, with another +301 event if removal would still be better; a DFE optimization on that chain is compared the same way (DFE taps re-extracted on the CTLE-less link).

> **Tip:** Chains containing EO/Fiber/OE also support equalizer optimization; the target function switches to optical metrics (TDECQ / optical eye opening) automatically.

11.6 IBIS / IBIS-AMI Device Models

The IBIS node brings vendor IBIS models (up to IBIS 8.0, ratified 2025-12) into the Expert pipeline: the [Model] I-V tables, [Ramp]/waveform tables, C_comp and package description of the .ibs file are linearised into a transmitter (open-circuit swing, output resistance, 20-80% edge) or receiver (input resistance) analog front end; when the .ami declares Ts4file that 4-port Touchstone (Tx_Port_Order/Rx_Port_Order 13-24 or 12-34) is the analog buffer instead, with Tx_V/Tx_R/Rx_R entering the termination and Usage Dep values resolved by AMI_Resolve. The AMI executable referenced by [Algorithmic Model] (Windows 64-bit .dll + .ami parameter file) is run per the IBIS 8.0 Chapter 10 reference flows. The package follows the IBIS 7.2-style precedence (node "Package source"): [Interconnect Model Group] Touchstone models or IBIS-ISS subcircuits (File_TS / File_IBIS-ISS, incl. external .ims files, differential pairs reduced to Sdd; "Interconnect model group" picks the group when a component declares several) > [Package Model]/[Define Package Model] ([Number Of Sections] stubs cascaded section by section as distributed lines with Fork branches as shunt loads; [Model Data] RLC matrices reduced to the pair's odd mode L–M, C+2Cm) > [Pin] > [Package]. A model carrying a [C Comp Model] (IBIS 7.0+ §6.1) uses that Touchstone / IBIS-ISS network as the die capacitance (TX: Driving, RX: Non-Driving; a Buffer_I probe terminal becomes the receiver input), "C_comp source" falls back to the lumped C_comp. IBIS-ISS (IBIS Interconnect SPICE Subcircuit 1.0) is solved by a built-in linear evaluator: R/L/C/K/V, ideal T lines, coupled lossy W lines (RLGC and frequency-table models), S-element Touchstone files, E/F/G/H controlled sources, nested subcircuits and .param expressions.

Source (highest precedence first)	Data	REA handling
[Interconnect Model Group]	File_TS Touchstone or File_IBIS-ISS subcircuit (this file or .ims)	differential pair reduced to Sdd, other ports per Unused_port_termination; IBIS-ISS solved by the built-in evaluator; "Interconnect model group" picks among several groups
[Package Model] / [Define Package Model]	[Model Data] R/L/C matrices (Full / Banded / Sparse) or [Number Of Sections] stubs	matrices reduced to the pair's odd mode L–M, C+2Cm; stubs cascaded section by section as distributed lines, Fork branches as shunt loads
[Pin] / [Package]	R_pin/L_pin/C_pin or R_pkg/L_pkg/C_pkg	lumped series R–L + shunt C

Source (highest precedence first)	Data	REA handling
[C Comp Model] (die capacitance)	File_TS typ/min/max or File_IBIS-ISS per corner + Param	TX uses Driving, RX Non-Driving / All; a Buffer_I terminal is the receiver input; not applied under a Ts4file front end; "C_comp source" falls back to the lumped C_comp

Placement: a side=TX IBIS node sits after the Source (optionally after a TX FFE) and before the channel nodes; a side=RX node sits after the channel and before the probe; an RX node directly followed by a TX node forms a repeater pair (IBIS §10.8 Retimer/Redriver, cascadable) and the chain must end in an Eye Probe. After choosing the .ibs file the [Model] and pin lists are filtered by side, the .ami/.dll are filled in from [Algorithmic Model], and the dialog shows the linearised values and reserved-parameter summary; the "AMI parameters" table edits Usage In/InOut parameters (Range rows are clamped, List rows use a drop-down, Corner rows follow the selected corner).

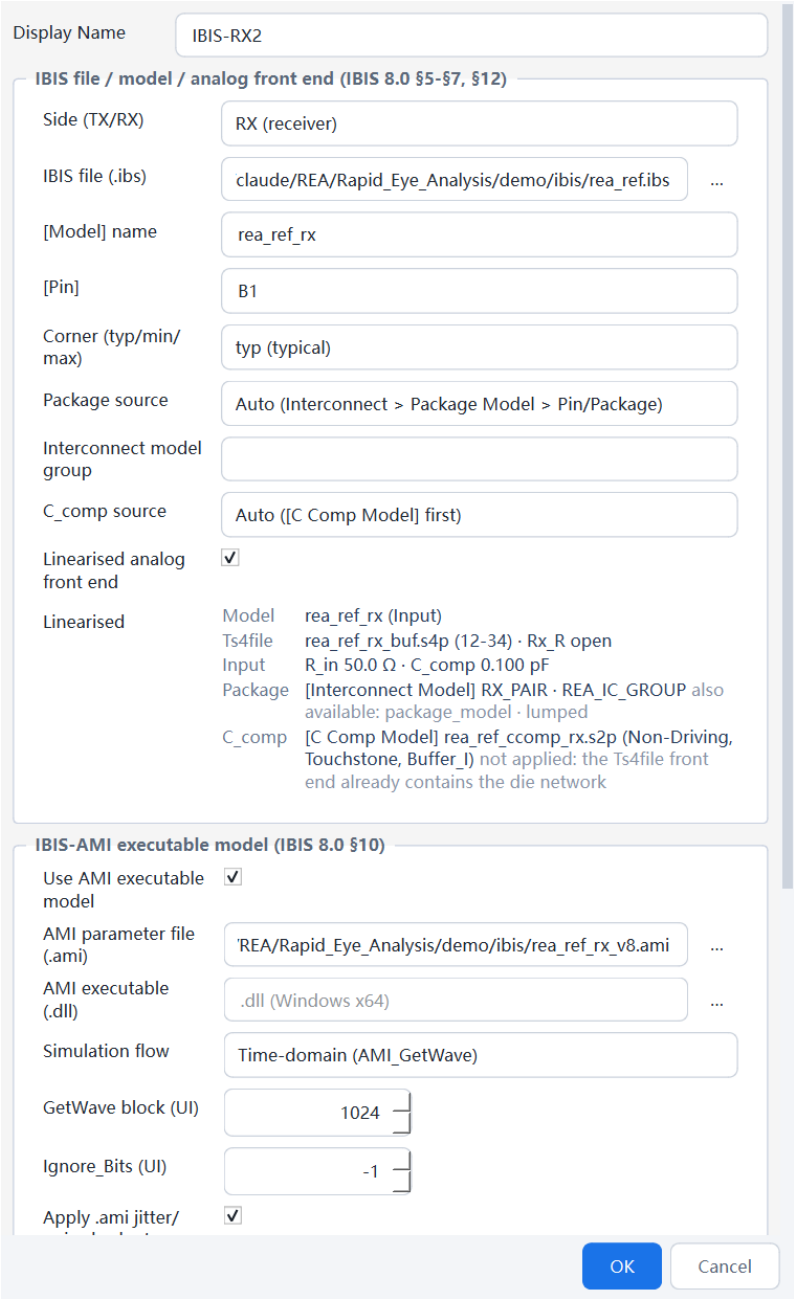

Figure 11-3 IBIS node dialog (RX example): file / model / pin / package source, linearised summary, AMI executable and 8.0 feature summary

Only Usage In / InOut parameters are editable; rows equal to their default are not written as overrides. Range rows are clamped to [min ... max]; List/Corner rows use the drop-down.

Branch	Parameter	Usage	Type	Format / range	Value	Description
Reserved	AMI_Version	Info	String	Value 8.0	8.0	Written again...
Reserved	Init_Returns_Impulse	Info	Boolean	Value True	True	AMI_Init ...
Reserved	GetWave_Exists	Info	Boolean	Value True	True	AMI_GetWave...
Reserved	Ignore_Bits	Info	Integer	Value 200	200	Maximum UI ...
Reserved	Max_Init_Aggressors	Info	Integer	Value 2	2	Up to two ...
Reserved	Model_Name	In	String	Value placeholder	placeholder	IBIS [Model] ...
Reserved	DLL_Path	In	String	Value placeholder	placeholder	Directory of t...
Reserved	DLL_ID	In	String	Value placeholder	placeholder	Unique ...
Reserved	Supporting_Files	Info	String	Table	rea_ref_rx_support.txt	Files the mod...
Reserved	Modulation_Levels	In	Integer	List 2 | 4	2	PAMn modulation ...
Reserved	PAM_Thresholds	Out	String	Value placeholder	placeholder	n-1 slicer ...

Reset to defaults OK Cancel

Figure 11-4 "AMI parameters" table: Usage In/InOut rows editable, Info/Out rows read-only

Flow	Meaning
Auto	time-domain when the .ami declares GetWave_Exists, else statistical
Statistical (AMI_Init)	the model equalises the channel impulse response; REA folds the eye from it (LTI)
Time-domain (AMI_GetWave)	the +/-0.5 V digital stimulus is fed block-wise; the model returns the decision-point waveform and, for RX models, clock_times used to sample the eye

Further conventions: Ignore_Bits (-1 = from the .ami) are dropped before folding; "Apply .ami jitter/noise budgets" puts Tx_Rj/Dj/Sj/DCD on the stimulus edges, Rx_Clock_Recovery_* on the sampling clock (only without model clocks) and Rx_Noise on the decision-point waveform, Table-format Tx_Jitter / Rx_Clock_PDF probability tables are sampled directly; a TX with GetWave paired with an Init-only RX follows step 6d — "Auto" prefers method 2 (one extra unit-impulse column passed to RX AMI_Init returns the RX filter itself; needs RX AMI_Version ≥ 7.2 and a free Max_Init_Aggressors slot), else method 1 (deconvolution). The §10.5 data-management parameters DLL_Path / DLL_ID are filled in by REA (executable directory, unique instance name) and missing Supporting_Files are reported. The results pane lists the flow, clock source, linearised values and the AMI output parameters/messages; over SCPI use FETC:PIPE:IBIS? for the node summary and the ibis_flow / rx_clock_src / ami_rx_out fields of FETC:PIPE:EYE?. The ibis sub-folder of demo ships a reference kit (rea_ref.ibs plus AMI transmitter/receiver and their IBIS 8.0 variants) to try the block immediately.

TX GetWave + RX Init combination (step 6d)	What happens
Auto (default)	method 2 when the RX AMI_Version is ≥ 7.2 and a Max_Init_Aggressors slot is free, else method 1
Deconvolve (method 1)	the RX EQ filter is deconvolved from the RX AMI_Init output and input (regularised; may under-estimate the boost where the channel loss is high)
Unit-impulse column (method 2)	one extra unit-impulse aggressor column is passed to RX AMI_Init; its output column is the exact RX gain/EQ filter (Figure 11-8)
Ignore TX GetWave	the TX GetWave output is discarded and the Init results are convolved as in step 6c

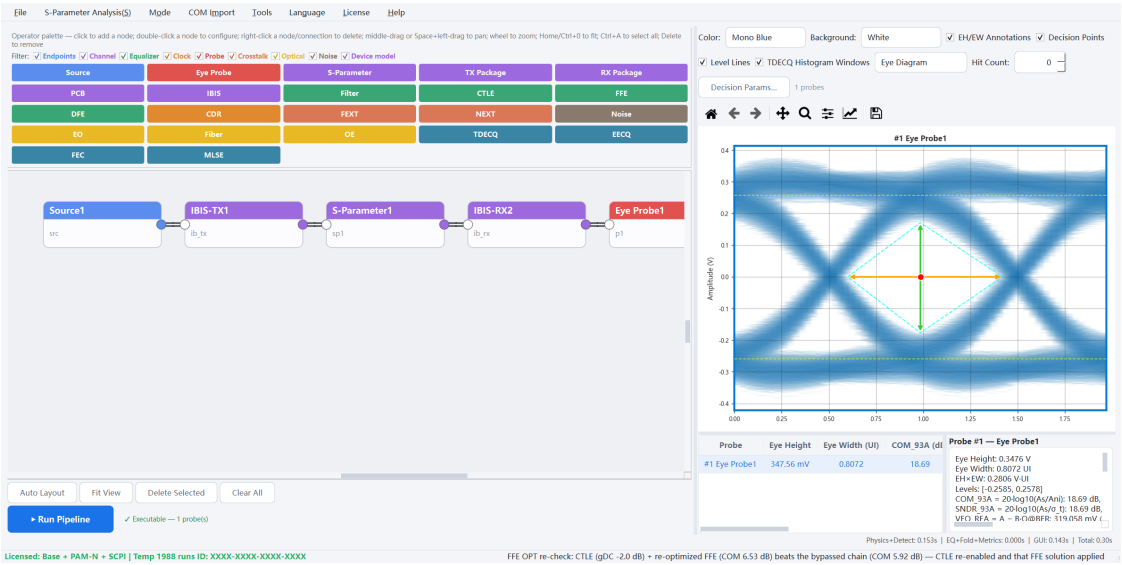

Figure 11-5 IBIS-AMI link: Source → IBIS(TX) → channel → IBIS(RX) → probe, time-domain GetWave eye and the IBIS section of the results pane

IBIS 8.0 / SerDes extensions: crosstalk injectors attached to the channel become AMI_Init aggressor columns (up to Max_Init_Aggressors per side; the statistical crosstalk uses the RX-equalised columns); when the RX model reports Adaptation_Valid the eye accumulates from the block after it first becomes true (Ignore_Bits is the cap); Rx_Decision_Time seeds the statistical sampling; PAMn via Modulation_Levels/PAM_Thresholds/PAM_Offsets. Repeaters: a Retimer samples the repeater RX clock_times, decides and regenerates the digital stimulus for the repeater TX, a Redriver passes the RX analog waveform on and the downstream TX AMI_Init input follows Tx_Impulse_Input. BCI link training (§10.9): when both .ami files declare the same BCI_Protocol the tool sets BCI_State=Training, ping-pongs AMI_Impulse until the terminal RX returns Converged (Impulse mode) and/or watches BCI_State in the GetWave block loop (node "BCI link training" can switch it off). [AMI Test Configuration] (§10.11) runs

over SCPI FETC:PIPE:IBIS:TEST? and is compared with the vendor golden data.

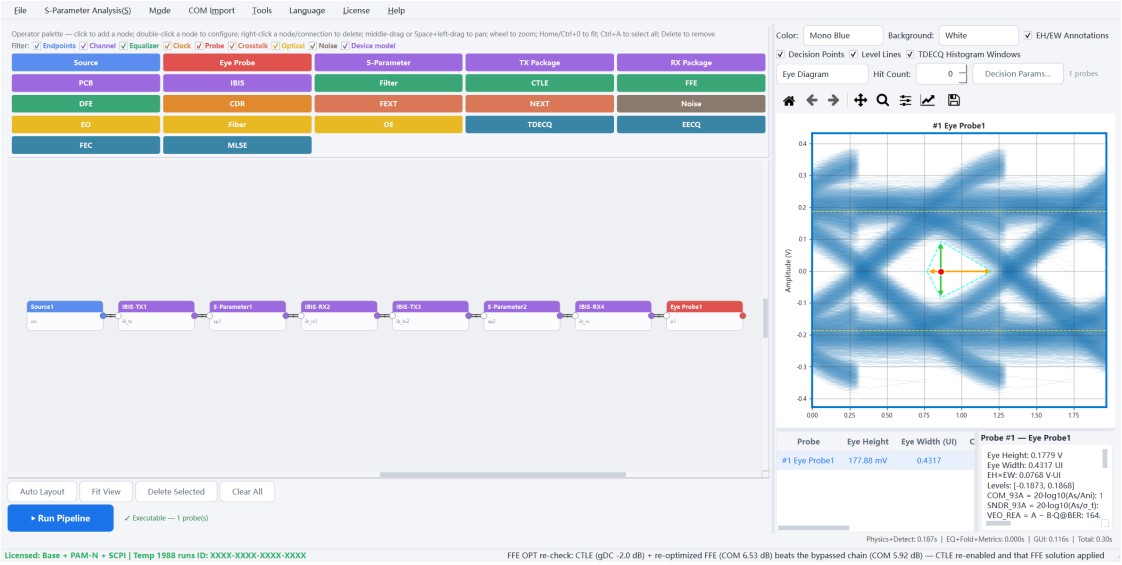

Figure 11-6 Repeater chain: an IBIS(RX) directly followed by an IBIS(TX) forms a Retimer pair; the downstream eye is driven by the regenerated stimulus

```
IBIS / IBIS-AMI: Flow time_domain/6a; Sampling clock: AMI clock_times (model CDR) (7751 clk)
  Ignore_Bits 200 UI; A_eq 1.000 V (ibis); T_r 12.6 ps (ibis:waveform); R_out 25.0 Ω / R_in 50.0 Ω
  PAM_Thresholds -0.126/0/0.126
  TX model rea_ref_tx (Output)
    V_oc 1.000 V, R_out 25.0 Ω, C_comp 0.120 pF, T_r 12.6 ps (waveform)
    package [Interconnect Model] TX_PAIR_ISS (IBIS-ISS)
    C_comp [C Comp Model] rea_ref_ccomp.iss:rea_ccomp_tx (Driving, IBIS-ISS)
    AMI rea_ref_tx.ami + rea_ref_tx.dll (v7.2, Init_Returns_Impulse True, GetWave_Exists True, GetWave used, DLL_ID REA_38824_ib_tx_tx)
    AMI output parameters: mode=auto
    AMI messages: rea_ref_tx: FFE [0 0.9 -0.1] swing 1 spu 32 rows 13984 aggressors 0
  RX model rea_ref_rx (Input)
    R_in 50.0 Ω, C_comp 0.100 pF
    package [Interconnect Model] RX_PAIR_ISS (IBIS-ISS)
    C_comp [C Comp Model] rea_ref_ccomp_rx.s2p (Non-Driving, Touchstone, Buffer_I)
    AMI rea_ref_rx.ami + rea_ref_rx.dll (v7.2, Init_Returns_Impulse True, GetWave_Exists True, GetWave used, DLL_ID REA_38824_ib_rx_rx)
    AMI output parameters: Rx_Noise=0.002, Rx_Decision_Time=143.562
      PAM4_UpperThreshold=0.126102, PAM4_CenterThreshold=0, PAM4_LowerThreshold=-0.126102
      PAM_Thresholds=-0.126102 0 0.126102, PAM_Offsets=0 0 0, Adaptation_Valid=True
      support_file=none, gain_trim_db=0, instance_id=, dfe_tap1=0.358
    AMI messages: rea_ref_rx: CTLE 3 dB, DFE 1 tap (adaptive), CDR bw 0.002 UI, NRZ, spu 32, aggressors 0, decision time 143.562 UI
  Jitter/noise budget: Rx_Noise σ 2.00 mV, uniform ±0.00 mV, sensitivity 10.0 mV
```

Figure 11-7 IBIS-ISS subcircuit packages (interconnect model group REA_IC_GROUP_ISS) and [C Comp Model] die networks: the IBIS section of the results pane lists the interconnect models (IBIS-ISS), the C_comp networks, DLL_ID and the AMI output parameters

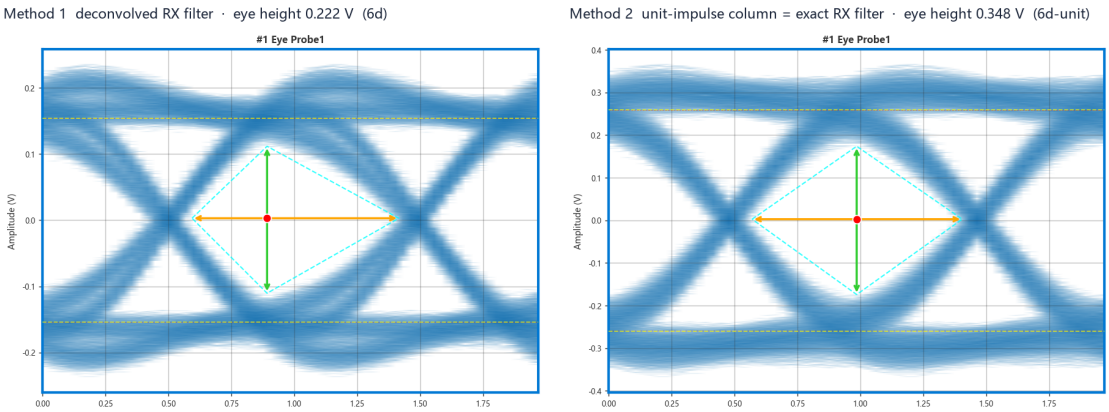

Figure 11-8 The two step-6d methods (TX GetWave + Init-only RX, same link): left = method 1 deconvolved RX filter, right = method 2 unit-impulse column giving the exact RX filter ("Auto" prefers it)

Note: AMI models are vendor black-box DLLs and run in the isolated compute process: a model crash only fails that probe, the application keeps running. Only 64-bit Windows models load; the block targets high-speed Ethernet / PCIe SerDes and leaves the DDR-style content out: non-linear I-V/V-T transient drivers, [Submodel]/[Driver Schedule] and Rx_Use_Clock_Input dual-input clock receivers; EBD/EMD board- and module-level files and [External Model] multi-lingual buffer models are not read.

Chapter 12 Optical Link Simulation and TDECQ

Add EO (electrical-to-optical modulator), Fiber, and OE (optical-to-electrical detector) nodes on the Expert canvas to simulate a complete optical interconnect: the electrical drive becomes an optical power waveform, propagates through fiber dispersion/attenuation, and is recovered by a photodetector. A TDECQ node evaluates transmitter quality per IEEE 802.3.

12.1 Building an Optical Link

The simplest link:

1. Enter Expert mode and clear the canvas (or branch off an existing chain).
2. Click Source, EO, Fiber, OE, Eye Probe in the palette to add five nodes.
3. Wire them in signal order: Source -> EO -> Fiber -> OE -> Probe.
4. Double-click nodes to set parameters (12.2), or keep defaults.
5. Click "Run Pipeline"; the probe shows the electrical eye after the whole optical link.

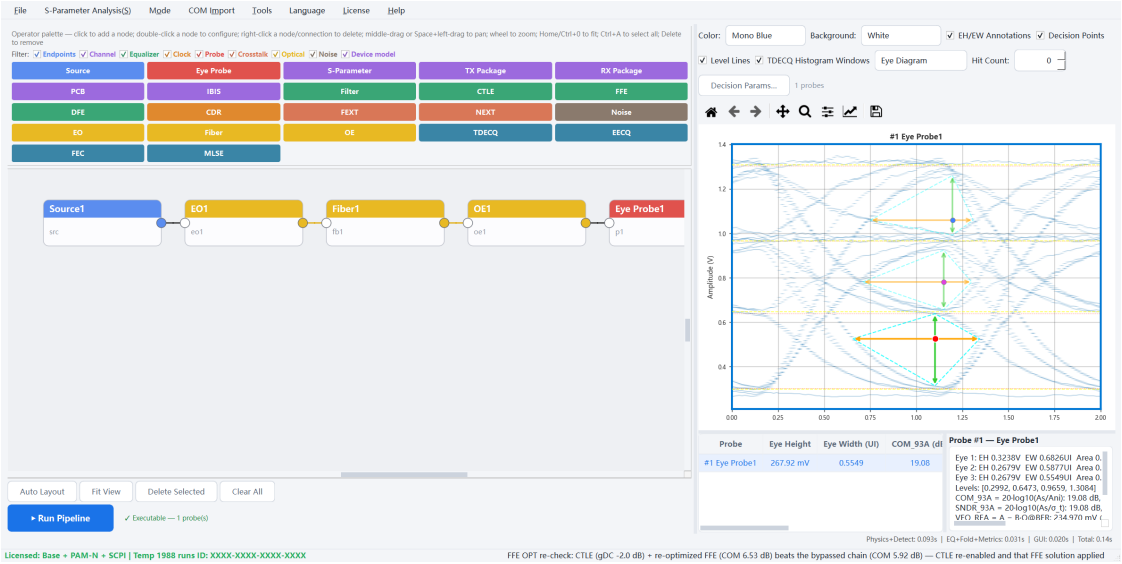

Figure 12-1 Optical link: Source -> EO -> Fiber -> OE -> probe

> **Tip:** The probe "signal domain" defaults to auto: after OE it shows an electrical eye; after EO/Fiber it shows an optical power eye. It can also be forced to electrical/optical.

12.2 Optical Block Parameters

Node	Key parameters
EO	Modulator type (MZM / EML / DML - switching applies that type default set), wavelength, extinction ratio ER, drive swing, EO bandwidth, laser power, RIN
Fiber	Fiber type (SMF / MMF), length (km), attenuation (dB/km), dispersion D (ps/nm/km) and slope, zero-dispersion wavelength, PMD coefficient
OE	Detector type (PIN / APD), responsivity R (A/W), PD/TIA bandwidth, transimpedance, dark current, TIA noise, baud-rate preset (one click applies a full parameter set)

> **Tip:** EO and OE both offer baud-rate presets (26/53/106 GBd etc.); EO/OE can also load a measured S21 response file (Custom mode). A specified response file that cannot be read or processed reports an error and stops that calculation. It is not replaced with an analytic model. When no file is specified, the selected analytic model is used.

> **Note:** Fiber wavelength should match the EO wavelength (the software auto-syncs and warns). Dispersion grows away from the zero-dispersion point and dominates at long reach + high rate.

12.3 TDECQ Evaluation

TDECQ (Transmitter and Dispersion Eye Closure Quaternary) is the IEEE 802.3 metric for PAM4 optical transmitters: a reference receiver (reference photodetection + 4th-order Bessel-Thomson filter + reference FFE) measures the optical eye closure in dB - lower is better.

Usage: add a TDECQ node and connect it directly after EO or Fiber (its input is fixed to optical). After running, the node shows the TDECQ (dB) result, and the probe table lists it too.

> **Note:** A TDECQ node must receive an optical signal - connect it after EO or Fiber, never after OE (the TDECQ reference receiver contains its own O/E conversion).

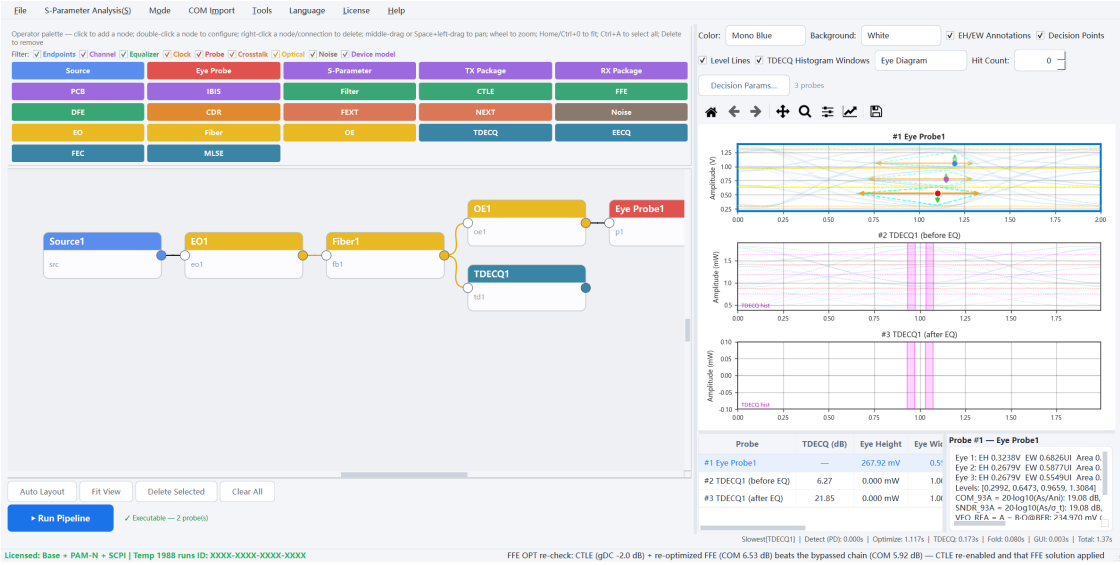

Figure 12-2 TDECQ node in parallel with an electrical-eye probe after the fiber

12.3.1 Main TDECQ Settings

Setting	Description
Spec preset	802.3 clause presets (121/178/184 and 802.3dj draft) - applies FFE tap count / DFE permission / thresholds
Baud rate	Default auto (derived from Source data rate)
Reference equalizer	FFE taps and spacing per selected clause; some clauses allow a 1-2 tap DFE

Setting	Description
Eye output	Show "filtered (unequalized)", "equalized", or both; the before-EQ eye is measured on its own with an identity equalizer (reference-receiver filter only, Ceq = 1, its own histogram windows and sampling phase, no compliance verdict); the equalized TDECQ should not exceed it
Details panel	The multi-eye details panel opens with a value-first summary line (TDECQ value / limit verdict / Ceq / clause), then the DFE tap, the FFE taps four per row, the formula and compliance rows; eye state and levels come last, so the default layout shows everything without scrolling (EECQ block alike; other probes keep their order); with TDECQ / EECQ probes the comparison table shows a "TDECQ (dB)" / "EECQ (dB)" column right after the probe column
Optimize button	Auto-tunes upstream EO drive etc. to minimize TDECQ

Tip: Optical chains are evaluated with TDECQ / optical eye opening rather than COM; an empty COM column for optical probes is normal.

12.3.2 Joint FFE + DFE solution (introduced for 802.3dj 200G/lane optics, extended to every reference receiver with a DFE)

The TDECQ reference equalizer of 802.3dj D3.1 (Clause 180 / 182 / 183) is a fixed 15-tap T-spaced FFE followed by a single decision-feedback tap; the feedback is modeled as correct and the coefficients are bounded by Table 180-16 (b(1) between 0 and 0.33, per-tap ratio limits relative to the main tap, |w(1)/w(0) - b(1) - w(-1)/w(0)| no more than 0.25, and so on). Post-cursor interference removed by the FFE costs noise enhancement (Ceq); the same interference removed by the DFE costs nothing. The standard therefore asks for a joint solution of the FFE and DFE coefficients at a given sampling phase and noise level, and accepts any alternative method that reports an equal or lower TDECQ.

Every TDECQ / EECQ measurement block (presets with a DFE as well as FFE-only presets) adds a refinement pass with the standard primary method after the regular equalizer optimization: the reference sequence is the test pattern (the known symbol sequence of the source, i.e. x(n) of the standard model; the decisions of the current equalized eye are used only when the block has no pattern), the 15 FFE taps and b(1) are solved together at a given noise level under the standard criterion with every Table 180-16 limit enforced inside the solver (FFE-only presets have no feedback term and the pre/post-cursor coupling reduces to |w(1)/w(0) - w(-1)/w(0)| no more than 0.25), the number of pre-cursor taps (0 to 3) and a grid of b(1) over 0 to 0.33 are scanned jointly for the global optimum, b(1) is then refined by golden section at the chosen number of pre-cursor taps, and noise and taps are re-solved iteratively

(the "iterative process" of the standard) until no further improvement. Every point is judged by the real TDECQ / EECQ and the previous result is replaced only when strictly better. When the eye is closed the decisions of the current equalized eye no longer follow the pattern, and the fit to those decisions is a different, reference-free equalizer solution; it then enters as a second candidate judged by the same criterion. The DFE decision phase is chosen jointly as well: each candidate phase is scored on the equalized eye after feedback, which avoids the mis-pick that a phase taken from the unequalized eye produces under strong feedback. The SCPI result carries two extra fields, opt_winner (source of the final choice, dj-joint / joint when the refinement won, dj-mmse / mmse for FFE-only presets) and joint (accepted or not / evaluations / wall time / metric before refinement).

The same idea is applied to the optical-chain FFE node optimized jointly with a DFE node (Expert mode): after the existing joint search finishes, the first DFE tap is re-searched with a grid plus golden-section pass, the FFE is re-solved at every point, and each candidate is judged by the real eye quality of the link; the previous result is replaced only when strictly better. Electrical joint optimization keeps its existing algorithm.

> **Tip:** The joint refinement leaves the original optimization objective untouched; it only moves the final judgement to the real figure of the link and accepts strictly better results only. The result is compared with the candidate before this refinement under the same configuration. Presets without a DFE use the same primary-method refinement. After changing presets or bounds, regenerate the eye and check the reported result.

Chapter 13 SCPI Remote Control

Rapid Eye Analysis supports remote control via the SCPI (Standard Commands for Programmable Instruments) protocol, suitable for automated test scripts and system integration.

13.1 Starting the SCPI Server

The SCPI service is built into the main application and starts automatically with Rapid Eye Analysis — no additional component needs to be installed or launched. Use "Tools → Enable SCPI Remote Control" in the menu bar to disable/re-enable it at any time, and "Tools → SCPI Service Status" to view the current service address and number of supported commands.

> **Tip:** The SCPI server runs unconditionally at application launch (even before the SCPI option is activated). Without the SCPI option, you can still remotely query the Host ID, check status, activate license keys, and read the error queue; however, "execution" commands such as INIT, MEAS, all OPT optimizers, and COM import return error -200,"SCPI license not authorized; activate via LIC:ACT:SCPI". Activating the permanent SCPI option key releases all commands immediately, no restart required.

Clients connect via telnet or socket on port 5025 (default). The protocol is SCPI-raw over TCP/IP with LF (0x0A) line terminator.

13.2 Command Reference

Supports IEEE 488.2 common commands and the following SCPI subsystems:

Command	Description
*IDN?	Query instrument identification
*RST	Reset all configuration
*OPC?	Wait for current operation to complete and return 1
*WAI	Wait for current operation to complete
*CLS	Clear status register and error queue
*OPT?	Query licensed options (e.g. BASE,PAM,SCPI,FEC)
*TST?	Self-test (internal deps only; no-file is not a failure)

Command	Description
ABORt	Abort eye computation and all optimizations (deliverable even while a long optimization blocks; includes SCPI-triggered FFE / RX FFE / CTLE optimizations, parameters unchanged after a stop)
CONF:FILE "path"	Set S-parameter file path
CONF:FILE:ADD "path"	Append file to cascade (mix s2p/s3p/s4p any order)
CONF:FILE:LIST?	List of loaded / cascaded files
CONF:S2P:DIR 0|1	S2P segment direction (0=1→2, 1=2→1 mirror)
CONF:S3P:PAIR <n>,<0-2>	n-th S3P file diff pair (0=(1,2)/1=(2,3)/2=(1,3))
CONF:S3P:DIR <n>,<0|1>	n-th S3P file direction (0=SE→diff, 1=diff→SE)
CONF:SIGN:PATT:FILE "path"	Custom pattern file (0/1 text; sets PATT=CUSTOM)
CONF:SIGN:TYPE NRZ|PAM4|...	Set signal type
CONF:SIGN:RATE 28.0	Set data rate (Gbaud)
CONF:SIGN:VOLT 1.0	Set signal amplitude (Vpp)
CONF:SIGN:SPU 64	Set samples per UI
CONF:SIGN:PATT PRBS7|SSPR|SSPRQ	Set pattern (SSPR locks NRZ, SSPRQ locks PAM4)
CONF:SIGN:SKIP 100|AUTO	Set skip symbol count
CONF:SIGN:RISE 0.2	Set TX rise time (UI), range 0.00–0.50
CONF:SIGN:FALL 0.2	Set TX fall time (UI), range 0.00–0.50
CONF:SIGN:GRAY ON|OFF	Enable/disable Gray Code (PAM-4/PAM-8 only)
CONF:SIGN:NOIS0 0.010	Set noise injection N■ amplitude (Vrms), before TX FFE
CONF:SIGN:NOIS1 0.010	Set noise injection N■ amplitude (Vrms), before Channel
CONF:SIGN:NOIS2 0.010	Set noise injection N■ amplitude (Vrms), before RX Filter
CONF:SIGN:NOIS3 0.010	Set noise injection N■ amplitude (Vrms), before CTLE
CONF:SIGN:NOIS4 0.010	Set noise injection N■ amplitude (Vrms), before RX FFE

Command	Description
CONF:SIGN:NOIS5 0.010	Set noise injection N■ amplitude (Vrms), before DFE
CONF:PORT:TX 0	Set transmit port index
CONF:PORT:RX 1	Set receive port index
CONF:FFE:ENAB ON|OFF	TX FFE enable/disable
CONF:FFE:PRE 1	Pre-cursor tap count
CONF:FFE:POST 2	Post-cursor tap count
CONF:FFE:SPAC 1.0	Tap spacing (UI)
CONF:FFE:TAPS 0.0,1.0,0.0,0.0	Set FFE tap coefficients
CONF:FFE:OPT	Auto-optimize FFE tap coefficients
CONF:FFE:PRESET <name>	TX FFE protocol preset (PCIe P0..Q9 / USB4-Pn / USB4v2-Pn)
CONF:FFE:UNIT_MAIN ON|OFF	FFE main tap normalize to 1.0
CONF:FFE:BMIN/BMAX <list>|NONE	TX FFE per-tap lower / upper bounds (written by presets or COM import; optimization stays within; ? reads back)
CONF:FFE:MAIN_AUTO ON|OFF	Main-tap auto: re-split pre / post around the largest |tap| after optimization
CONF:FFE:OPTJOINT	Joint TX FFE + DFE optimization
CONF:RXF:ENAB ON|OFF	RX filter enable/disable
CONF:RXF:TYPE GAUSSIAN|...	Set filter type
CONF:RXF:BAND 20.0	Set filter -3dB bandwidth (GHz)
CONF:RXF:RISE 50.0	Set Gaussian filter rise time (ps), bidirectionally linked with bandwidth
CONF:CTLE:ENAB ON|OFF	CTLE equalizer enable/disable
CONF:CTLE:PRES <index>	Set CTLE preset by index (0=Custom, 1=OIF-28G-VSR, 2=OIF-56G-PAM4, ...)
CONF:CTLE:MODE 2POLE|3POLE|3P2Z	CTLE mode (2-Pole / 3-Pole / 3-Pole-2-Zero)

Command	Description
CONF:CTLE:GAIN -6.0	CTLE DC gain gDC (dB)
CONF:CTLE:GAIN2 -1.0	CTLE low-frequency gain gDC2 (dB, 3-Pole)
CONF:CTLE:ZERO 8.0	CTLE zero frequency fz (GHz)
CONF:CTLE:POLE1 15.6	CTLE pole 1 frequency fp1 (GHz)
CONF:CTLE:POLE2 14.1	CTLE pole 2 frequency fp2 (GHz)
CONF:CTLE:FLF 1.2	CTLE low-frequency pole fLF (GHz, 3-Pole)
CONF:CTLE:G1 -3.5	CTLE gain g1 (dB, 3P2Z mode)
CONF:CTLE:G2 -1.5	CTLE gain g2 (dB, 3P2Z mode)
CONF:CTLE:FZ1 5.0	CTLE zero 1 frequency fz1 (GHz, 3P2Z mode)
CONF:CTLE:FZ2 12.0	CTLE zero 2 frequency fz2 (GHz, 3P2Z mode)
CONF:CTLE:FP3 20.0	CTLE pole 3 frequency fp3 (GHz, 3P2Z mode)
CONF:CTLE:OPT	Auto-optimize CTLE gDC/gDC2 (2D grid search)
CONF:CTLE:GAIN:MIN/MAX <dB>	CTLE gDC optimization lower/upper bound (honored by optimizer)
CONF:CTLE:GAIN2:MIN/MAX <dB>	CTLE gDC2 optimization lower/upper bound (3-Pole)
CONF:RXFFE:ENAB ON|OFF	RX FFE enable/disable
CONF:RXFFE:PRE 1	RX FFE pre-cursor tap count
CONF:RXFFE:POST 2	RX FFE post-cursor tap count
CONF:RXFFE:SPAC 1.0	RX FFE tap spacing (UI)
CONF:RXFFE:TAPS 0.0,1.0,0.0,0.0	Set RX FFE tap coefficients
CONF:RXFFE:OPT	Auto-optimize RX FFE tap coefficients
CONF:RXFFE:OPTJ	Jointly optimize RX FFE + DFE tap coefficients
CONF:RXFFE:PRESET PCIE7	RX FFE protocol preset (PCIe 128 GT/s 29-tap)

Command	Description
CONF:RXFFE:UNIT_MAIN ON|OFF	RX FFE main tap normalize to 1.0
CONF:RXFFE:BMIN/BMAX <list>|NONE	RX FFE per-tap lower / upper bounds (presets or COM import; ? reads back)
CONF:RXFFE:MAIN_AUTO ON|OFF	RX FFE main-tap auto pre / post re-split
CONF:RXFFE:IND	RX FFE custom tap indices (sample units)
CONF:RXFFE:FLOA:ENAB ON|OFF	Enable/disable floating FFE (auto-searches far-end reflection positions for tap groups)
CONF:RXFFE:FLOA:ENAB?	Query floating FFE status
CONF:RXFFE:FLOA:BMAXG <v>	Set floating FFE bmaxg max tap magnitude (0.01-1.0, default 0.2)
CONF:RXFFE:FLOA:BMAXG?	Query bmaxg
CONF:RXFFE:FLOA:GRO <n>	Set N_bg floating groups (1/2/3)
CONF:RXFFE:FLOA:GRO?	Query N_bg
CONF:RXFFE:FLOA:TPG <n>	Set N_bf taps per group (1-12)
CONF:RXFFE:FLOA:TPG?	Query N_bf
CONF:RXFFE:FLOA:SEAR <n>	Set N_bmax search range (20-200 UI)
CONF:RXFFE:FLOA:SEAR?	Query N_bmax
CONF:RXFFE:FLOA:OPT	Run floating FFE reflection detection + tap extraction
CONF:RXFFE:FLOA:DFE ON|OFF	Enable/disable floating DFE
CONF:RXFFE:FLOA:DFE?	Query floating DFE status
CONF:RXFFE:FLOA:DFE:BMAXG <v>	Floating DFE bmaxg (default 0.05)
CONF:RXFFE:FLOA:DFE:BMAXG?	Query

Command	Description
CONF:RXFFE:FLOA:DFE:GRO <n>	Floating DFE N_bg
CONF:RXFFE:FLOA:DFE:GRO?	Query
CONF:RXFFE:FLOA:DFE:TPG <n>	Floating DFE N_bf
CONF:RXFFE:FLOA:DFE:TPG?	Query
CONF:RXFFE:FLOA:DFE:SEAR <n>	Floating DFE N_bmax
CONF:RXFFE:FLOA:DFE:SEAR ?	Query
CONF:RXFFE:FLOA:DFE:DELT <v>	DFE DAC quantization step (alias of CONF:DFE:STEP, shared param)
CONF:RXFFE:FLOA:DFE:DELT ?	Query
CONF:RXFFE:FLOA:DFE:OPT	Run floating DFE reflection detection + tap extraction (consistent with the COM reference implementation)
CONF:RXFFE:FLOA:CTL <n> / :RSS:MAX <v> / :TAIL:STAR <n>	COM floating-tap parameters RXFFE_FLOAT_CTL / B_float_RSS_MAX / N_tail_start (set by import; ? reads back)
CONF:DFE:ENAB ON|OFF	DFE equalizer enable/disable
CONF:DFE:NTAP 4	DFE tap count (1-32)
CONF:DFE:TAPS 0.1,0.05,0.02,0.01	DFE tap coefficients (comma-separated)
CONF:DFE:OPT	Auto-optimize DFE taps (supports DAC quantization step)
CONF:DFE:PRESET <name>	DFE protocol preset (PCIE3..7 / USB3.2-Gen2 / USB4-Gen2/Gen3)
CONF:DFE:POS	DFE floating tap positions (UI offsets)
CONF:DFE:STEP <v>	DFE DAC quantization step (matches the COM DFE quantization setting, shared fixed+floating)

Command	Description
CONF:DFE:STEP?	Query
CONF:DFE:BMIN/BMAX <list>|NONE	DFE per-tap lower / upper bounds (b_min / b_max; ? reads back)
CONF:DFE:IMPL?	Whether the current DFE is the implicit 1-tap derived by COM import (1/0)
CONF:NOIS:DM:FREQ0 100.0	DM sinusoidal noise frequency for slot 0 (MHz)
CONF:NOIS:DM:AMP0 10.0	DM sinusoidal noise amplitude for slot 0 (mVpp)
CONF:NOIS:CM:FREQ0 100.0	CM sinusoidal noise frequency for slot 0 (MHz)
CONF:NOIS:CM:AMP0 10.0	CM sinusoidal noise amplitude for slot 0 (mVpp)
CONF:NOIS:CM:CMRR 40.0	Common-mode rejection ratio (dB)
CONF:JITT:RJ 0.010	Random jitter RMS (UI)
CONF:JITT:SJ 100:0.1,...	Sinusoidal jitter pairs (freq_MHz:amp_UI, comma-separated; NONE clears)
CONF:JITT:SSC:ENAB ON|OFF	Jitter SSC enable/disable
CONF:JITT:SSC:FREQ 33.0	Jitter SSC frequency (kHz)
CONF:JITT:SSC:PPM 5000	Jitter SSC deviation (ppm)
CONF:JITT:SSC:MODE DOWN|CENTER	Jitter SSC mode
CONF:CDR:ENAB ON|OFF	CDR enable/disable
CONF:CDR:TAPP POST|PRE	CDR tap point (after full EQ / after CTLE)
CONF:CDR:PDTY MM|BB	CDR phase detector type
CONF:CDR:ORD 1|2	CDR loop order
CONF:CDR:BAND 10.0	CDR loop bandwidth (MHz)
CONF:CDR:DAMP 0.70	CDR damping factor
CONF:CDR:KP 0.001	CDR Kp manual override
CONF:CDR:KI 0.0001	CDR Ki manual override

Command	Description
CONF:CDR:PDG 1.0	CDR PD gain
CONF:CDR:CLIP 2.0	CDR phase clipping (UI)
CONF:CDR:PHAS 0.0	CDR initial phase (UI)
CONF:CDR:FREQ 0.0	CDR initial frequency offset
CONF:CDR:DEL 0	CDR lock delay (symbols)
CONF:SSC:ENAB ON|OFF	TX SSC enable/disable
CONF:SSC:FREQ 33.0	TX SSC frequency (kHz)
CONF:SSC:PPM 5000	TX SSC deviation (ppm)
CONF:SSC:MODE DOWN|CENTER	TX SSC mode
CONF:LEV:ENAB ON|OFF	Level ratio enable/disable
CONF:LEV:RAT 1.0,1.0,...	Level ratios (comma-separated)
CONF:FLEV:ENAB ON|OFF	Force levels enable/disable
CONF:FLEV:VAL 0.25,0.75,...	Force level values (comma-separated)
CONF:MEAS:DEC:T 0.10	Decision-point time search radius (UI)
CONF:MEAS:DEC:V 0.50	Decision-point voltage search range scale
CONF:DIFF:ENAB ON|OFF	Differential mode enable/disable
CONF:DIFF:PAIR 0|1|2|3	Port pairing: 0=(0,1)(2,3) 1=(0,2)(1,3) 2=(0,3)(1,2) 3=custom (see also PAIR AUTO)
CONF:DIFF:P1P/P1N/P2P/P2N <n>	Custom pairing (PAIR 3) port numbers
CONF:DIFF:PAIR AUTO	Guess the S4P differential pairing and apply it (single 4-port file)
CONF:DIFF:PAIR:AUTO?	Pairing guess: index,scores of the 3 schemes,gap,cand=candidates (no change)
CONF:MEAS:HIT 100	Hit count threshold for eye metrics
CONF:MEAS:VADJ 0.0	Vertical adjustment for decision levels (V)

Command	Description
CONF:MEAS:TADJ 0.0	Timing adjustment for sampling point (UI)
CONF:MEAS:TCEN:LO 0.0	Timing center low bound (UI)
CONF:MEAS:TCEN:HI 1.0	Timing center high bound (UI)
CONF:FEC:ENAB ON|OFF	Enable / disable pre- and post-FEC BER analysis (needs the FEC option or remaining runs)
CONF:FEC:ENAB?	Query the FEC analysis switch
CONF:MLSE:ENAB ON|OFF	Measured MLSE is for preview validation only; not available in the released product (see 15.8)
CONF:MLSE:ENAB?	Query the MLSE measurement switch
CONF:MLSE:B1 |DFE	Partial-response coefficient b1: DFE = first DFE tap; a number = manual (|b1| <= 1)
CONF:MLSE:B1?	Query the b1 source / value
CONF:MLSE:TRAC 16..4096	Viterbi traceback depth (UI), default 128
CONF:MLSE:TRAC?	Query the traceback depth
CONF:FEC:PRES ""	FEC standard preset (one of seven, see Chapter 15)
CONF:FEC:PRES?	Query the active preset
CONF:FEC:LANes 1|2|4|8	Physical lane count
CONF:FEC:LANes?	Query the lane count
CONF:FEC:BADDed |DEF	Added BER (DEF = the default specified for the preset)
CONF:FEC:BADDed?	Query the added BER
CONF:COM:IMPORT "csv" "xlsx" [PEAK|PULSE]	COM import (CSV + Excel config)
CONF:COM:IMPORT:MLSERx ON|OFF	COM import: for CSVs with MLSE on, adopt the receiver solution COM itself deployed (RxFFE leaves post-cursor 1 to the 1+b1·D trellis, implicit 1-tap DFE); default ON, see 15.8
CONF:COM:IMPORT:MLSERx?	Query the switch

Command	Description
CONF:COM:IMPORT (no args)	Import using the saved CSV / Excel / S4P paths
CONF:COM:IMPORT:CSV/XLSX /S4P "path"	Set an import path alone (saved, no import triggered; ? reads back)
CONF:PKG:ENAB ON|OFF	Package model enable/disable
CONF:PKG:GAMMA0/A1/A2 <val>	Transmission line propagation params
CONF:PKG:TAU <val>	Propagation delay τ
CONF:PKG:Z0 <val>	Reference impedance Z■ (Ω)
CONF:PKG:RD:TX/RX <val>	Driver/receiver impedance R_d (Ω)
CONF:PKG:{CBUMP|CP|CV}:{TX|RX} <val>	Package capacitance params (pF)
CONF:PKG:ZP:TX/RX	Trace segment lengths z_p (mm)
CONF:PKG:ZP:NEXT/FEXT	NEXT/FEXT TX-side trace length (mm)
CONF:PKG:ZC	Segment impedances Z_c (Ω)
CONF:PKG:PCB:ENAB ON|OFF	PCB board trace enable/disable
CONF:PKG:PCB:{ZBP:TX|ZBP:RX|ZC:RX|TAU|GAMMA}	PCB trace parameters
CONF:PKG:PCB:ZC:TX/RX <val>	PCB impedance (TX/RX separate, Ω)
CONF:PKG:PCB:ZBP:{NEXT|FEXT} <mm>	NEXT/FEXT TX-side PCB length
CONF:TX:GAUS:ENAB ON|OFF	TX Gaussian rise-time filter enable
CONF:TX:GAUS:TR <ps>	TX rise time Tr (ps)
CONF:CHAN:INT COM|CUBIC	Interpolation mode (COM mag/phase or cubic)
CONF:CHAN:CAUS:AP ON|OFF	COM Alternating Projections causality (recommended)

Command	Description
CONF:CHAN:CAUS ON|OFF	Cepstral minimum-phase causality (strong reconstruction). Auto-enabled/reverted on S-parameter load based on measured causality health (SCPI event +303); manual setting or COM import overrides auto control until *RST
CONF:NOIS:ADD	COM A_DD data-dependent noise
CONF:NOIS:ETA0 <val>	COM η■ / DER target
CONF:NOIS:BER <val>	Target BER (e.g. 1e-4)
CONF:NOIS:SPECber <ber>	COM specBER (DER_0) decision error-rate target
CONF:NOIS:SIGMA_RJ <UI>	COM σ_RJ random jitter (UI rms)
CONF:NOIS:SNR_TX <dB>	COM TX signal-to-noise ratio (dB)
CONF:NOIS:SNR_TX:WC0 ON|OFF	COM SNR_TXwC0: divide σ_TX by the TX FFE main tap c0 (transmitter noise referenced to the un-de-emphasized swing; FETC:EYE:COM? reports SNR_TX_eff_dB / TX_c0)
CONF:NOIS:SIGMA_N <mV>	COM receiver thermal noise σ_N (mV)
CONF:NOIS:BUDGetmap ON|OFF	Map noise budget into displayed eye (default ON; η■/σ_N/A_DD/SNR_TX/σ_RJ derived-injected, display only, COM_93A unchanged, optimizers isolated)
CONF:SAMP:PHAS:METH <method>	Sample phase method
CONF:OPT:TYPE NM|GRID	Optimizer search algorithm (Nelder-Mead / grid search)
CONF:OPT:FFE:STEP <v>	Grid-search FFE tap step (0.005-0.05)
CONF:OPT:LOC:SEAR <n>	Grid-search local refinement span
CONF:OPT:FOM COM	Optimization objective (fixed to COM; EYE retired)
CONF:OPT:CTLE:STEP <dB>	CTLE gain optimization step size
CONF:OPT:PREC LOW|MED|HIGH	Optimizer sampling precision (session preference, unaffected by *RST)
CONF:OPT:COM:TOPN <n>	COM fast-evaluation top-N cursors (same source as the menu)

Command	Description
CONF:OPT:GRID:CAND <n>	Grid-search candidate cap per stage
CONF:OPT:MEM:MAX <gb>	Optimization memory cap (GB); local-search candidate cap derived per tap count
CONF:FFE|RXFFE|CTLE:OPT:ASYNc, CONF:FFE|RXFFE:OPTJ:ASYNc	Async optimization variants (return immediately, non-blocking)
FETC:OPT:STAT? / PROG? / ELAP?	Async optimization status idle|running|done|failed|stopped / progress % / elapsed s (ABORt-able)
CONF:TDR:WIND EXP|GAUS|...	Set TDR window function type
CONF:TDR:WIND:PAR 1.0	Set window function parameter (β/α)
CONF:TDR:PORT:ROW <n>	TDR port row (S-matrix row, 0-based)
CONF:TDR:PORT:COL <n>	TDR port column (S-matrix column, 0-based)
CONF:TDR:VEL 0.7	Set velocity factor
CONF:TDR:MODE / CAUS / REN / PEXT / EDGE:SMO / STEP:SMO	Headless TDR parameters: mode, causality, renormalization, port extension, edge / step smoothing (independent of the GUI TDR window)
CONF:TDR:DC:METH / DC:VAL / BAND:EXT / CORR:IMP / DISP:ONEWAY	DC extrapolation method / value, bandwidth extension, impedance correction, one-way display
CONF:TDR:FREQ:STAR/STOP, TIME:STAR/STOP, GATE:STAR/STOP/STAT, WIND:MODE/COEF	Frequency / time ranges, gating, window mode and coefficient
FETC:TDR:IMPU? / STEP?	TDR impulse / step response data
INIT:TDR	Execute TDR computation
FETC:TDR:IMP?	Query TDR impedance array
FETC:TDR:TIME?	Query TDR time axis
FETC:TDR:DIST?	Query TDR distance axis
INIT	Start eye diagram generation (async)

Command	Description
ABOR	Abort current computation
FETC:EYE:COUN?	Query number of eyes
FETC:EYE:HEIG? [n]	Query eye height (V)
FETC:EYE:WIDT? [n]	Query eye width (UI)
FETC:EYE:AREA? [n]	Query eye area (V*UI)
FETC:EYE:ALL?	Query all eye metrics
FETC:EYE:COM?	Query COM_93A + SNDR_93A noise budget (protocol BER-cond/-inv)
FETC:EYE:VEO?	Query VEO + SNR_REA + σ_eye (eye-statistics BER-inv)
FETC:EYE:CHAN?	Channel quality metrics (IL / fitted IL / ILD / FOM_ILD / RL / delay / PSXT / ICR / SDC21 / SCD21 / skew / h-1 h0 h+1 / ISI; positive-loss dB, see 10.17.5)
CONF:WAVE:SAVE "path"	Export the last eye's decision-point waveform capture as CSV (streaming / no capture → -221)
CONF:MASK:TYPE|EH|EW|EW Vref	Eye mask shape none|rect|diamond|hex / height V / width UI / reference-line width UI (see 10.12.1)
CONF:CHAN:SKEW txp,txn,rxp,rxn	Single-ended p/n skew (ps, COM 4.x Txpskew etc.; applied to 4-port channels before the mixed-mode transform); query form reads back
CONF:CHAN:KAPPa k1,k2	Differential reflection scaling kappa1/kappa2 (0 = reflections at tp0/tp5 omitted)
CONF:NOIS:ACCM vrms[,fmax]	AC common-mode noise AC_CM_RMS (V rms) and integration limit ACCM_MAX_Freq (Hz, 0 = fb)
CONF:CHAN:IRTRunc thr	COM impulse-response truncation threshold (fraction of the peak; 0 = none; the COM import applies the configured value, default 1e-3)
CONF:NOIS:ICN mV	CSV ICN Gaussian fallback: without aggressor files σ_XT = ICN·σ_x (filled by the COM import; ICN_src in FETC:EYE:COM?)

Command	Description
FETC:EYE:BATH?	Bathtub summary (EH@BER / EW@BER per BER + σ; target BER = specBER or 1e-12)
FETC:EYE:BATH:CURV? V|T[,n]	Bathtub curve data x:ber;…
FETC:EYE:MASK?	Mask margin PASS/FAIL, hits, top/bottom/left/right margins
FETC:FEC?	FEC result for the current Normal-mode eye (name=value)
FETC:PIPE:FEC?	FEC result of an Expert-mode fec node
FETC:MLSE?	Measured MLSE gain of the current Normal-mode eye (name=value, both conventions)
FETC:PIPE:MLSE?	MLSE result of an Expert-mode mlse node
FETC:RUNT?	Query run time (seconds)
MEAS:EYE:ALL?	One-shot measurement (INIT + wait + FETCH)
MEAS:EYE:HEIG? [n]	Trigger computation and return eye height
CONF:XTALK:ENAB ON|OFF	Enable/disable crosstalk injection
CONF:XTALK:ADD:FEXT "path" [volt]	Add FEXT crosstalk path (optional voltage Vpp)
CONF:XTALK:ADD:NEXT "path" [volt]	Add NEXT crosstalk path (optional voltage Vpp)
CONF:XTALK:ICN:AMPL a_ft,a_nt	Aggressor amplitudes A_ft/A_nt used for the ICN report only (V; 0 = same as each path; waveform unchanged)
CONF:XTALK:CLEA	Clear all crosstalk paths
CONF:XTALK:COUN?	Query crosstalk path count
CONF:XTALK:LIST?	Query all crosstalk paths
CONF:XTALK:PATH<n>:TXP <v>	Set path n TX port
CONF:XTALK:PATH<n>:RXP <v>	Set path n RX port
CONF:XTALK:PATH<n>:VOLT <v>	Set path n voltage (Vpp)

Command	Description
CONF:XTALK:PATH<n>:DELAY <v>	Set path n delay
CONF:XTALK:PATH<n>:{P1P|P1N|P2P|P2N} <n>	S4P diff port pairing (TX+/TX-/RX+/RX-)
CONF:XTALK:PATH<n>:DEL	Delete path n
MEAS:XTALK:ICN?	Query total ICN (mV)
MEAS:XTALK:FEXT?	Query FEXT ICN (mV)
MEAS:XTALK:NEXT?	Query NEXT ICN (mV)
SYST:ERR?	Query error queue (also SYST:ERR:ALL? / :COUN?); negative = error, +301 = optimization verdict advisory, +302 = optimizer event (no improvement / internal fallback / failure info, text follows the UI language)
SYST:ERR:NEXT? / ALL? / COUN?	Error queue: next / all / count
STAT:OPER|QUES:COND? / EVEN? / ENAB	IEEE 488.2 status registers (operation / questionable): condition, event, enable (also STAT:PRES)
*ESE / *ESR? / *SRE / *STB?	Standard event / status byte registers (IEEE 488.2)
SYST:VERS?	Query SCPI version
SYST:HELP:HEAD?	Query all supported commands
SYST:CONF:LOAD ""	Load an REA configuration file (same entry as File → Load config; a file with an Expert pipeline switches mode and rebuilds the canvas; S-parameter paths fall back to the same-named file next to the config)
SYST:CONF:SAVE ""	Save the displayed configuration, including Normal FEC/MLSE and the Expert pipeline; shares the File menu entry point
SYST:LANGuage zh_CN|en_US / SYST:LANG?	Switch or query GUI language immediately while retaining parameters and results

Command	Description
DISP:PAGE SIGNAL|MEASUREMENTS|RESULTS / DISP:PAGE?	Select or query the Normal control page
DISP:STAGe SIGNAL|TXFFE|FILTER|CTLE|CDR|RXFFE|DFE / DISP:STAG?	Select or query the signal-chain editor without changing module enable states
INIT:PAUSe ON|OFF / INIT:PAUS?	Pause, resume or query Normal eye generation using the same state as the GUI pause button
DISP:FEC:LIVE ON|OFF / DISP:FEC:LIVE?	Set or query live updates on the active mode's FEC result panel
DISP:FEC:PROBe "" / DISP:FEC:PROB?	Select or query an Expert FEC result probe
DISP:FEC:UPDate / DISP:FEC:CANCel	Update or cancel FEC result computation using the panel actions
DISP:FEC:PAUSe ON|OFF / DISP:FEC:PAUS?	Pause, resume or query FEC result updates
DISP:FEC:SAVE ""	Export the displayed FEC metrics and bins using the same path as Save CSV
SYST:GPU ON|OFF	Enable/disable CUDA acceleration (syncs the GUI menu check)
SYST:GPU?	Query GPU state (ON / OFF / OFF,UNAVAILABLE)
SYST:HW:CORE <n>	CPU core cap (0 = auto)
SYST:HW:RAM <gb>	RAM cap in GB (0 = auto)
SYST:HW:VRAM <gb>	VRAM cap in GB (0 = auto)
SYST:OOM:POL IGNORE|STOP	Default action on a memory warning (headless / no-dialog case)
SYST:FEED "<text>"	Send one feedback message (version / OS / Host ID / license attached automatically; no screenshots over SCPI)
SYST:FEED:CONT "<contact>"	Leave an email or WeChat ID for a reply

Command	Description
SYST:FEED:ONL?	Is the feedback service online (1/0) - offline is normal, probe before sending
SYST:FEED:REPL?	Fetch replies from the author / bot (JSON array, last 20)
LIC:HOST?	Query Host ID (used to generate keys via keygen)
LIC:BASE?	Query whether base permanent key is active (1/0)
LIC:PAM?	Query whether PAM-N option is active (1/0)
LIC:SCPI?	Query whether SCPI option is active (1/0)
LIC:FEC?	Whether the FEC option is active (1/0)
LIC:TRI?	Query trial/temp counts: remaining,total,bonus
LIC:STAT?	Query full license status (HOST/BASE/PAM/SCPI/REM/TOT/BON)
LIC:ACT:BASE "<key>"	Activate base permanent key
LIC:ACT:PAM "<key>"	Activate PAM-N option key
LIC:ACT:SCPI "<key>"	Activate SCPI option key (releases INIT/MEAS/OPT immediately)
LIC:ACT:FEC ""	Activate the FEC option (base license required)
LIC:ACT:TEMP "<key>"	Activate temporary key (does not unlock SCPI)
LIC:ACT:LAST?	Query last activation/deactivation result ("1,success_msg" or "0,error_msg")
LIC:DEAC:BASE	Deactivate base license (also clears all options)
LIC:DEAC:PAM	Deactivate only PAM-N option
LIC:DEAC:SCPI	Deactivate only SCPI option (note: current SCPI connection will close)
LIC:DEAC:FEC	Deactivate the FEC option only
LIC:DEAC:TEMP	Clear temporary bonus

13.2.2 Expert Pipeline and IBIS-AMI Commands

The Expert canvas (Chapter 11) is fully scriptable: add / remove nodes, connect them, set parameters, run, and read every probe. Node types and parameter names match the canvas dialogs (see each dialog's tooltips).

Command	Description
CONF:MODE NORMAL|EXPERT	Switch Normal / Expert; CONF:MODE:COPY copies the current Normal configuration onto the canvas
CONF:PIPE:CLEAR	Clear the canvas
CONF:PIPE:NODE:ADD <type>[,<id>]	Add a node (source/sparam/package_tx/package_rx/pcb/ffe/rx_filter/ctle/dfe/cdr/noise/fext/next/eo/fiber/oe/tdecq/eecq/ibis/fec/mlse/probe)
CONF:PIPE:NODE:DEL <id>	Remove a node
CONF:PIPE:NODE:PAR <id>,<key>,<value>	Set a node parameter (the value may contain commas, e.g. the ibis ami_params JSON)
CONF:PIPE:NODE:PAR? <id>,<key>	Query a node parameter
CONF:PIPE:CONN <from>,<to> / DISC	Connect / disconnect
CONF:PIPE:LAYOUT	Auto layout
CONF:PIPE:VAL?	Validate: OK or a |-separated ERROR/WARNING list
CONF:PIPE:LIST? / EDG?	Node list id:type;... / edges from->to;... (incl. crosstalk / noise injector edges)
CONF:PIPE:OPT <node>[,NM|GRID]	Legacy module-level optimization (auto-picks the downstream probe)
CONF:PIPE:OPT:PAUS ON|OFF (+?)	Pause / resume the running async Expert optimization
FETC:PIPE:OPT? / :STAT? / :PROG? / :ELAP?	Last Expert optimization result JSON / status / progress % / elapsed s
ABOR:PIPE:OPT	Cancel the running Expert optimization
CONF:PIPE:RUN	Run the pipeline (blocking; wait with *OPC?)
CONF:PIPE:RUN:ASYN	Run asynchronously using the GUI worker; control and query from the same TCP session

Command	Description
FETC:PIPE:RUN:STAT?	Asynchronous / GUI RUN state: idle, running, paused, done, failed or cancelled
CONF:PIPE:RUN:PAUS ON|OFF / :PAUS?	Pause, resume or query Expert RUN using its GUI controller; start remotely with RUN:ASYN
ABOR:PIPE:RUN	Stop GUI- or SCPI-started Expert RUN; abort commands take priority in the same session
CONF:PIPE:OPT:CTLE|FFE|DFE <probe>,<node>	Optimise one equaliser node on a probe chain
FETC:PIPE:PROB?	Probe list
FETC:PIPE:EYE? <probe>	Probe eye metrics (eye_height/eye_width/com_db/…; IBIS chains add ibis_flow/rx_clock_src/ibis_links/ami_rx_out …)
FETC:PIPE:EYE:REP? <probe>	The probe's results-pane text (same renderer as the GUI pane, lines joined by |)
FETC:PIPE:CHAN? <probe>	Channel quality metrics of the probe chain (same convention as FETC:EYE:CHAN?; optical probes report available=0)
CONF:PIPE:WAVE:SAVE <probe>,"path"	Export the probe's decision-point waveform capture as CSV (same as the multi-eye right-click)
FETC:PIPE:BATH? <probe> / :BATH:CURV? <probe>,V|T[,n] / :MASK? <probe>	Probe bathtub summary / curve data / mask margin (same convention as FETC:EYE:*; the mask uses the global CONF:MASK settings)
FETC:PIPE:MASK? <probe>	Mask-margin result of the probe
FETC:PIPE:IBIS? <ibis node>	IBIS node summary: model / linearised values / package source and detail (package_detail=matrix:coupled|sections:N, interconnect_kind=ts|iss) / [C Comp Model] (c_comp_model=ts|iss|lumped) / .ami reserved parameters / jitter tables / Supporting_Files / DLL exports / spec check / test configurations (nothing executed)

Command	Description
FETC:PIPE:IBIS:TEST? <ibis node>[,<config>]	Run an [AMI Test Configuration] of the .ibs and compare with the golden data (ok=1|0 plus per-item errors)
FETC:PIPE:FEC? <fec node>	FEC pre/post BER
FETC:PIPE:MLSE? <mlse node>	Measured MLSE detector gain (COM-comparable / real-DFE conventions)

Common IBIS node parameters: side (tx|rx), file_path (.ibs), model_name, pin, corner, package_source (auto|lumped|package_model|interconnect), use_analog, use_ami, ami_file_path, dll_file_path, flow (auto|init|getwave), ami_params (JSON), getwave_block_ui, ignore_bits, use_ami_jitter, tx_getwave_rx_init (auto|deconvolve|unit_impulse|skip_tx_getwave), interconnect_group, c_comp_source (auto|lumped), repeater_type (auto|redriver|retimer), bci_training (auto|off|on). The IBIS fields of FETC:PIPE:EYE? also carry ibis_tx_pkg_detail / ibis_tx_ccomp / ami_tx_dll_id / ibis_jitter_table (same for rx). Example:

CONF:MODE EXPERT

CONF:PIPE:CLEAR

CONF:PIPE:NODE:ADD source,src ; CONF:PIPE:NODE:ADD ibis,ib_tx ; CONF:PIPE:NODE:ADD sparam,sp1

CONF:PIPE:NODE:ADD ibis,ib_rx ; CONF:PIPE:NODE:ADD probe,pr1

CONF:PIPE:CONN src,ib_tx ; CONF:PIPE:CONN ib_tx,sp1 ; CONF:PIPE:CONN sp1,ib_rx ; CONF:PIPE:CONN ib_rx,pr1

CONF:PIPE:NODE:PAR sp1,file_path,D:/models/channel.s4p

CONF:PIPE:NODE:PAR ib_tx,side,tx ; CONF:PIPE:NODE:PAR ib_tx,file_path,D:/models/vendor.ibs

CONF:PIPE:NODE:PAR ib_rx,side,rx ; CONF:PIPE:NODE:PAR ib_rx,file_path,D:/models/vendor.ibs

CONF:PIPE:NODE:PAR ib_rx,ami_params,{"ctle_gain_db": 6}

CONF:PIPE:NODE:PAR ib_tx,package_source,interconnect ; CONF:PIPE:NODE:PAR ib_tx,interconnect_group,REA_IC_GROUP_ISS

CONF:PIPE:NODE:PAR ib_rx,c_comp_source,auto ; CONF:PIPE:NODE:PAR ib_tx,tx_getwave_rx_init,auto

FETC:PIPE:IBIS? ib_rx ; CONF:PIPE:VAL? ; CONF:PIPE:RUN ; *OPC? ; FETC:PIPE:EYE? pr1

13.2.1 First-Time Activation on a Fresh Machine

On a machine where the SCPI option has not yet been activated, the entire authorization flow can be completed via SCPI alone (no GUI required):

*IDN? # Verify connection

LIC:HOST? # Get this machine's Host ID

(Generate base / SCPI keys from Host ID using the keygen tool)

LIC:ACT:BASE "<complete base license key>" # Activate base

LIC:ACT:SCPI "<complete SCPI option key>" # Then activate SCPI option

LIC:STAT? # Verify all activations

INIT # INIT/MEAS/OPT now released

> **Tip:** On activation failure: existing keys are preserved (already-active keys are not cleared); the error is pushed to the SYST:ERR? queue. LIC:ACT:LAST? returns the detailed success/failure message. Option keys (PAM/RE/SCPI/TEMP) require the base key to be active first.

13.3 S-Parameter Analysis Window Control

The following commands remotely open, configure, and close S-parameter analysis windows. Each window has a unique integer ID and multiple windows of the same type can be open simultaneously. SCPI-opened windows do not restore previous manual settings, ensuring a clean state.

13.3.1 Window Management

Command	Description
WIND:OPEN CASC|CONV|DEEM|SPLI|TDR|SMIT	Open S-param analysis window, returns ID
WIND:OPEN?	Query current open window type
WIND:CLOSe <id>	Close specified window
WIND:LIST? [type]	List all windows (optionally filter by type)

13.3.2 Format Converter

Command	Description
CONF:CONV:FILE <id>,"path"	Load S-parameter file
CONF:CONV:PORT:ENAB <id>,ON|OFF	Port extraction enable/disable

Command	Description
CONF:CONV:PORT:PERM <id>,1,3,2,4	Set port permutation order
CONF:CONV:DIFF:ENAB <id>,ON|OFF	Differential conversion enable/disable
CONF:CONV:DIFF:PAIR <id>,SEQ|CROSS13|...	Differential port pairing mode
CONF:CONV:DIFF:P1P/P1N/P2P/P2N <id>,n	Differential port indices
CONF:CONV:DIFF:OUTP <id>,SDD|SDC|...	Differential output mode
INIT:CONV <id>	Execute format conversion
CONF:CONV:SAVE <id>,"path"	Save converted result

13.3.3 Cascade

Command	Description
CONF:CASC:FILE:ADD <id>,"path"	Add S-param file to cascade list
CONF:CASC:FILE:CLEar <id>	Clear file list
CONF:CASC:FILE:LIST? <id>	Query current file list
CONF:CASC:PORT:ENAB <id>,ON|OFF	Port mapping enable/disable
CONF:CASC:PORT:IN1/IN2/OUT1/OUT2 <id>,n	Set WCM port mapping
CONF:CASC:S2Port:IN/OUT <id>,1|2	S2P segment direction in cascade tool (IN=2 → mirror)
CONF:CASC:DIFF:ENAB <id>,ON|OFF	Differential conversion enable/disable
CONF:CASC:DIFF:PAIR/P1P/.../OUTP	Differential parameters (same as CONV)
INIT:CASC <id>	Execute cascade computation

Command	Description
CONF:CASC:SAVE <id>,"path"	Save cascade result

13.3.4 De-embed

Command	Description
CONF:DEEM:DUT <id>,"path"	Load DUT (device under test) file
CONF:DEEM:LEFT:ENAB <id>,ON|OFF	Left fixture enable/disable
CONF:DEEM:LEFT:FILE <id>,"path"	Load left fixture file
CONF:DEEM:RIGH:ENAB <id>,ON|OFF	Right fixture enable/disable
CONF:DEEM:RIGH:FILE <id>,"path"	Load right fixture file
CONF:DEEM:PORT:ENAB <id>,ON|OFF	Port mapping enable/disable
CONF:DEEM:PORT:IN1/IN2/OUT1/OUT2 <id>,n	Set WCM port mapping
CONF:DEEM:DIFF:ENAB/PAIR/P1P/.../OUTP	Differential parameters (same as CONV)
INIT:DEEM <id>	Execute de-embedding
CONF:DEEM:SAVE <id>,"path"	Save de-embedded result

13.3.5 Splitter

Command	Description
CONF:SPLI:FILE <id>,"path"	Load S-parameter file
CONF:SPLI:NSEG <id>,n	Set number of segments
CONF:SPLI:RAT <id>,r1,r2,...	Set segment ratios (comma-separated)
CONF:SPLI:SMOO:ENAB <id>,ON|OFF	Smoothing filter enable/disable

Command	Description
CONF:SPLI:SMOO:WIND <id>,n	Smoothing window size
CONF:SPLI:PORT:ENAB <id>,ON|OFF	Port mapping enable/disable
CONF:SPLI:PORT:IN1/IN2/OUT1/OUT2 <id>,n	Set WCM port mapping
CONF:SPLI:DIFF:ENAB/PAIR/.../OUTP	Differential parameters (same as CONV)
INIT:SPLI <id>	Execute split computation
FETC:SPLI:COUN? <id>	Query number of result segments
CONF:SPLI:SAVE <id>,idx,"path"	Save a single segment result
CONF:SPLI:SAVE:ALL <id>,"dir"	Batch save all segments to directory

13.3.6 GUI TDR

Command	Description
CONF:GTDR:FILE <id>,"path"	Load S-parameter file
CONF:GTDR:WIND <id>,kaiser|...	Set window function type
CONF:GTDR:WIND:PAR <id>,value	Set window function parameter
CONF:GTDR:VEL <id>,0.65	Set velocity factor
CONF:GTDR:PORT / CAUS / REN / PEXT / DISP / EDGE:SMO / GATE:STAT/STAR/STOP/TYPE	Port, causality, renormalization, port extension, display mode, edge smoothing, gating (all take <id>)
CONF:GTDR:DIFF:ENAB/PAIR/OUTP	Differential parameters
INIT:GTDR <id>	Execute TDR computation
CONF:GTDR:SAVE <id>,"path.csv"	Save TDR result as CSV

13.3.7 Smith Chart

Command	Description
CONF:SMIT:FILE:ADD <id>,"path"	Add S-parameter file
CONF:SMIT:FILE:CLEar <id>	Clear file list
CONF:SMIT:FILE:LIST? <id>	Query file list
CONF:SMIT:PARAM <id>,S11|S21|...	Set displayed parameter
CONF:SMIT:CHARt:TYPE <id>,Z|Y	Chart type (Z impedance / Y admittance)
CONF:SMIT:ZREF <id>,<ohm>	Reference impedance (50/75/100Ω etc.)
CONF:SMIT:VSWR <id>,ON|OFF	VSWR contour circles display
CONF:SMIT:DIFF:ENAB/PAIR/OUTP	Differential parameters
INIT:SMIT <id>	Plot Smith chart

13.3.8 S-Parameter Workflow Example

Example: 2-port cascade

1. WIND:OPEN CASC — Open cascade window, returns ID (e.g. 1)
2. CONF:CASC:FILE:ADD 1,"file1.s2p" — Add first file
3. CONF:CASC:FILE:ADD 1,"file2.s2p" — Add second file
4. INIT:CASC 1 — Execute cascade
5. CONF:CASC:SAVE 1,"result.s2p" — Save result
6. WIND:CLOS 1 — Close window

> **Tip:** *RST closes all SCPI-opened S-parameter analysis windows.

13.4 Eye Diagram Typical Workflow

> **Tip:** The bundled demo/scpi_example.txt contains this full command sequence (step-by-step EQ and one-click COM import); replace <DEMO> with the demo folder path and send line by line.

The following is an example Python automation script workflow:

1. Connect to the SCPI server (socket connect)
2. CONF:FILE "channel.s4p" — Load S-parameter file
3. CONF:SIGN:TYPE PAM4 — Set signal type
4. CONF:SIGN:RATE 56.0 — Set data rate
5. CONF:FFE:ENAB ON; CONF:FFE:OPT — Enable and optimize FFE
6. MEAS:EYE:ALL? — One-shot measurement, returns all eye metrics
7. FETC:EYE:HEIG? 1 — Query eye height of the 1st eye

> **Tip:** FETC:EYE:HEIG?, FETC:EYE:WIDT?, and FETC:EYE:AREA? accept an optional number n to query the nth eye (1-based). Without a parameter, all eye values are returned (comma-separated).

> **Note:** The SCPI server operates in single-client mode (only one connection at a time). For multi-client concurrent access, implement your own proxy/queuing mechanism.

Chapter 14 FAQ and Troubleshooting

Q: Eye diagram generation fails with "Insufficient S-parameter bandwidth"

A: Eye height is affected by S-parameter bandwidth, which must be significantly higher than the data rate. Use an S-parameter file that covers a wider frequency range, or reduce the data rate setting.

Q: Trial uses have been exhausted

A: Contact your vendor to obtain a license key. View your Host ID in "License → License Management" and provide it to your vendor. A temporary key immediately adds run counts and unlocks all features.

Q: PAM-N signal type is not selectable

A: Once trial counts are exhausted, PAM-N eye diagrams require the PAM-N option or a temporary license. Activate a PAM-N option key or temporary key in "License → License Management".

Q: CUDA acceleration toggle is unavailable

A: Both of the following must be satisfied: (1) CuPy is installed (pip install "cupy-cuda12x[ctk]==14.0.1"); (2) the system has an NVIDIA GPU. CUDA acceleration is free for all users and independent of licensing. Check "Tools -> CUDA Acceleration Status" for detailed diagnostics.

Q: Cascading/de-embedding results look abnormal

A: Verify that the S-parameter files have the correct port count and port ordering. Incorrect cascading order or mismatched port definitions will cause erroneous results.

Q: 4-port splitting shows small bumps at certain frequencies

A: Amplitude bumps of <2% may appear at resonance frequencies where eigenvalue phases pass through odd multiples of π. This is a physical resonance characteristic, not a bug. Two-segment cascading reconstruction accuracy is ~1e-13.

Q: S-parameter analysis window is unresponsive

A: Long-running tasks (e.g., high-order PRBS eye diagram) use asynchronous threads and should not block the UI. If the UI becomes unresponsive, it may be due to insufficient memory. Try reducing the PRBS order or closing other windows.

Q: How to view the user manual

A: Click menu bar "Help" → "User Manual" to automatically generate and open the PDF manual. First-time generation requires the reportlab package (pip install reportlab).

> **Tip:** If you encounter an issue not covered in this manual, check the status bar and console output for detailed information.

Chapter 15 Pre- and Post-FEC BER Analysis

Every high-speed link relies on FEC (forward error correction) to bring the raw physical-layer error rate down to what the system can tolerate. So "is the eye good enough" ultimately becomes "what is the error rate before correction, and what is left after it". This chapter covers the FEC analysis shown directly below the eye diagram, which reports both numbers and judges them against the relevant IEEE 802.3 / PCIe specification.

FEC analysis is an **option**, on the same footing as PAM-N and SCPI (see Chapter 4); it is usable during the trial period.

15.1 Enabling It

1. Produce an eye diagram first (Chapters 3-7). FEC results are derived from the decision-point statistics of that eye.

2. Open the left "Measurements / FEC" page and tick "FEC analysis" in its FEC group.

   → GUI feedback: Set the coding preset, physical lanes and added BER in the FEC group

3. Pick the preset matching the link under test (15.2).

4. A results strip appears below the eye: four metric cards on top, an error-distribution histogram underneath.

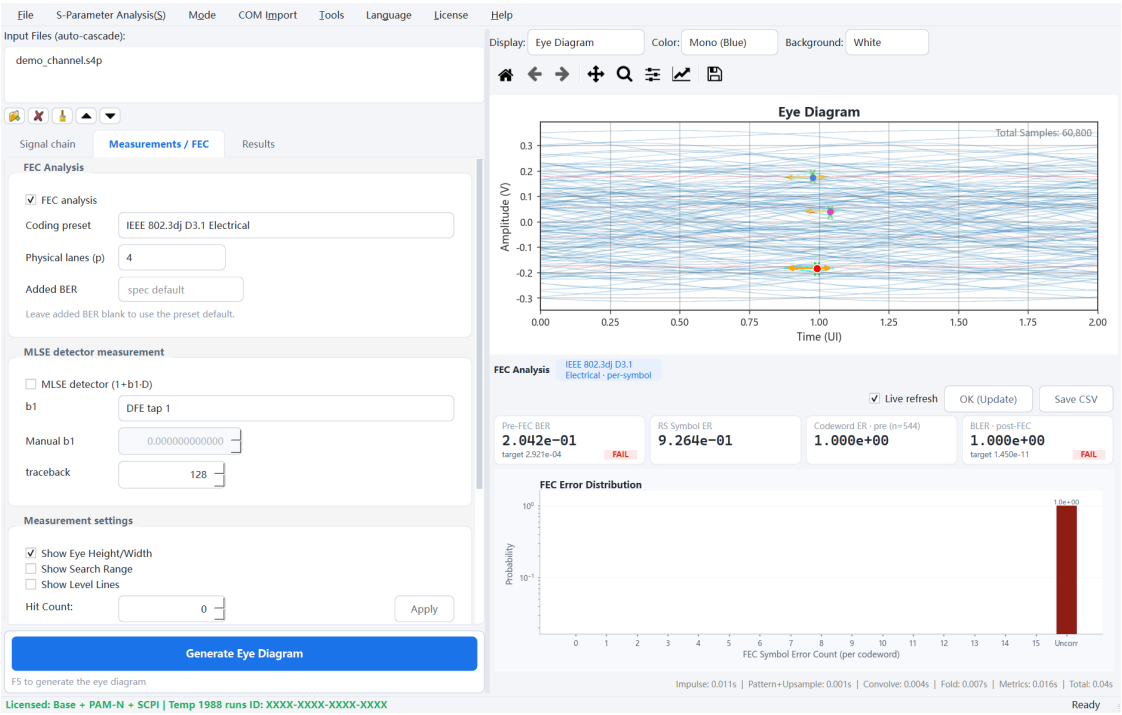

Figure 15-1 FEC results strip below the eye

> **Tip:** Drag the decision point on the eye and the FEC result follows, so you can see directly how the threshold choice affects the pre-FEC error rate. For very long patterns, clear "Live refresh" and use the "OK / Update" button instead.

15.2 Standard Presets

The preset selects both the FEC parameters and the pass/fail criteria. Choosing the wrong preset is not an error, but the verdict becomes meaningless - pick the standard the link actually uses:

Preset	Code	Modulation	Typical use
IEEE 802.3dj D3.1 Electrical	RS(544,514)	PAM4	200G/lane electrical
IEEE 802.3ck 100G/lane	RS(544,514)	PAM4	100G/lane electrical
IEEE 802.3bs Clause 119	RS(544,514)	PAM4	200G/400G
IEEE 802.3 Clause 91 KR4	RS(528,514)	NRZ	100GBASE-KR4/CR4
IEEE 802.3 Clause 74 BASE-R	(2112,2080)	NRZ	10G/25G-KR/CR
PCIe 6/7/8 FLIT	FLIT FEC	PAM4	PCIe 6.0/7.0/8.0
IEEE 802.3dj IM-DD Optical Inner FEC	Hamming(128,120)	PAM4	800G/1.6T DR/FR/LR optical inner code

Set **lanes** (1/2/4/8) to the physical lane count of the link; it determines how Ethernet test blocks are formed. The **Clause 74** preset applies to NRZ links only - selecting it with PAM4 reports an explicit error rather than a meaningless number. The **IM-DD Optical Inner FEC** preset reports the pre-inner BER (measured) and post-inner BER (t=1 guaranteed-bound model, limit 2.281e-4, RS(544,514) outer code not chained). Optical chains require the inline TDECQ topology (tdecq->...->probe->fec); the measurement point is the eye after the TDECQ reference receiver equalizer. Electrical chains reuse the per-symbol machinery.

15.2.1 BERadded

BERadded is the "added error rate": during a receiver interference tolerance test the instrument injects a specified amount of impairment, and that contribution has to be accounted for. Leave it empty to use the default specified for the preset (0 for the dj preset - the same basis as the target values that specification itself quotes); enter 1.6e-5 for the 802.3dj 178.2 receiver interference tolerance test.

15.3 Reading the Results

For Ethernet (RS-FEC) presets the four cards are:

- **Pre-FEC BER** - the bit error ratio before correction, i.e. the raw link performance. The target for the selected standard is printed below it, with a PASS / FAIL verdict.

- **RS Symbol ER** - RS symbol error ratio before correction.
- **Codeword ER pre** - probability that a codeword contains at least one error before correction.
- **BLER post-FEC** - the codeword error ratio **after** correction, i.e. the fraction the FEC cannot repair. This is the actual Ethernet criterion (802.3dj requires below 1.45e-11).

The Clause 74 preset shows Pre-FEC BER / block ER pre / uncorrectable block ratio / post-FEC BER instead. PCIe presets show FBER / byte error ratio / FLIT error ratio pre / FLIT error ratio post.

The **error-distribution histogram** below gives the probability of k errored symbols per codeword (x axis 0, 1, 2 ...; the rightmost bar is the part beyond the correction capability). With multi-lane interleaving the statistics follow the interleaved codeword the decoder actually sees - the same basis as the BLER criterion. Bars concentrated to the left mean plenty of margin; once the rightmost bar rises, uncorrectable codewords are already occurring.

> **Tip:** "Save CSV" exports the four card values together with the full distribution, which makes comparing several setups easy.

15.4 Patterns and Compute Time

Results are computed **symbol by symbol on the real pattern** rather than by plugging an average error rate into a formula - so changing the pattern (PRBS13 / PRBS31 / SSPRQ) changes the result, which is exactly what a real link does. Full periods from PRBS7 up to PRBS31 can be analysed (PRBS31 is on the order of a billion symbols); long patterns are accelerated on the GPU automatically (enable GPU under the Tools menu).

> **Note:** Very long patterns (PRBS27/PRBS31) take seconds to tens of seconds per run. The results strip shows progress and can be paused or cancelled at any time. Clear "Live refresh" and use "OK / Update" after adjusting the decision point.

15.4.1 How Jitter and Noise Enter the Statistics

Injected white noise and jitter (RJ / SJ / SSC) are reflected faithfully in the per-symbol statistics, so a desktop pre-check in the style of a receiver tolerance test - stress up step by step and read the pre- and post-FEC error rates - works directly.

Jitter is counted at the sampling phase **without clock-recovery tracking** by default - the most conservative reading: whatever you inject is what is counted. Enabling CDR gives the alternative reading: jitter inside the loop bandwidth is absorbed by tracking while jitter beyond it still counts in full, and the two can be compared directly. If the equalized eye cannot support clock recovery, the CDR disengages safely with a calibration-degraded notice (see 10.9.2).

15.5 Expert Mode

In Expert mode FEC is a node you attach after a probe: drag the "FEC" node from the palette and connect it to an Eye Probe. After running the pipeline the same results strip appears below the eye of that probe; with several probes, switch between them using the drop-down at the top left of the strip.

Node parameters match Normal mode (preset / lanes / BERadded).

15.6 SCPI Control

Normal mode:

```
CONF:FEC:ENAB ON
CONF:FEC:PRES "IEEE 802.3dj D3.1 Electrical"
CONF:FEC:LANes 4
CONF:FEC:BADDed DEF
INIT
*OPC?
FETC:FEC?
```

Expert mode (fec node):

```
CONF:PIPE:NODE:ADD fec,f1
CONF:PIPE:CONN p1,f1
CONF:PIPE:RUN
FETC:PIPE:FEC? f1
```

The query returns one line of "name=value" fields; the common ones (Ethernet presets) are:

Field	Meaning
status	OK means the result is valid; otherwise a status note (15.7)
preset / standard / lanes	Active preset, standard family, lane count
ser / pre_ber	Symbol / bit error ratio before correction
rsser / bins / bler	RS symbol ER / distribution / post-FEC codeword ER
target_pre_ber / target_post	The two targets of that standard
pass_pre / pass_post	The two verdicts (PASS / FAIL)
margin_pre_db / margin_post_db	Margin against the targets, in dB

Field	Meaning
coverage / n_rows	Pattern coverage / symbols included

PCIe presets use p_byte / flit_pre / flit_post instead; Clause 74 uses block_pre / bler / post_ber. Licensing and metering are covered in Chapter 4.

15.7 Status Messages and Troubleshooting

Message / symptom	Meaning and what to do
Insufficient data	Not enough samples or valid decisions for statistics - use a longer pattern, or check the decision point sits inside an open eye
Closed eye	No usable opening at the decision point - open the eye first (equalisation / lower rate)
Optical probes not supported	FEC analysis currently covers electrical probes only
No specified FEC mapping for this modulation	Preset and modulation do not match (Clause 74 is NRZ only)
All FAIL with the distribution piled at the right	Link errors far exceed what this FEC can correct - an expected result, not a software fault

15.8 MLSE Detector Measurement (measured gain)

Note: The measured MLSE module is not available in the released product. Normal startup hides its controls and does not permit mlse nodes; enable commands report that the feature is not available in this release. The instructions below apply to preview validation only. COM result import and MLSE flags in CSV files remain available.

When COM 4.x runs with OP.MLSE on, the delta-COM in its results is the **analytic** gain of Annex 178A: an estimate, from the noise statistics and an error-event union bound, of how much more noise margin an ideal MLSE has over the DFE slicer. It is not a waveform simulation. The REA MLSE block reports the same quantity **measured**: it takes the decision-point samples **before** the DFE from its own rendered time-domain waveform, runs a real Viterbi detector on the 1+b1·D partial-response trellis (b1 = first DFE tap, L states, traceback N_tc) and compares hard error counts with the DFE slicer on the very same

waveform.

Both conventions are reported:

- **COM-comparable** (genie): DFE tail taps b2..bN are cancelled with the true symbols and the slicer reference is an ideal DFE without error propagation - the same assumptions as COM DER_DFE / delta-COM, so the value can be compared directly with delta_COM in the CSV;
- **real DFE**: tail taps use each survivor path's own decisions (with error propagation) and the slicer reference is the real DFE - closer to a real receiver.

The gain definition matches delta-COM: the **actual noise realisation** extracted with the true symbols (residual ISI, jitter converted to amplitude, non-Gaussian parts included) is scaled as a whole and bisected to the target symbol error ratio (the source specBER, default 1e-4); the ratio of the scalings the two detectors tolerate is the noise-margin gain gain_margin_db = 20·log10(g_MLSE / g_slicer). The as-rendered hard-count SER and the Q-equivalent gain are reported as well. Nothing is approximated analytically.

How to use it in preview validation (not available in the released product):

- **Normal mode**: tick "MLSE detector (1+b1·D)" in the "Eye metrics" group on the left, choose the b1 source (DFE tap 1 / manual) and the traceback depth; after generating the eye the metrics panel shows two "MLSE (...)" lines. After a COM import whose CSV carries MLSE the same line ends with "[CSV delta-COM x.xx dB]".
- **Expert mode**: attach an **mlse** node after a probe (it may coexist with a fec node, one of each per probe); after RUN the result is appended to that probe's metrics text.
- SCPI: CONF:MLSE:ENAB / B1 / TRAC and FETC:MLSE? (Normal), FETC:PIPE:MLSE? <node> (Expert); the tokens carry b1 / b1_fit / n_syms / target_ser and the two groups genie_* / surv_* (ser_slicer, ser_mlse, gain_q_db, g_slicer, g_mlse, gain_margin_db, ok) plus csv_delta_com_db.

Notes: a DFE (or a manual b1) is required; the statistical threshold is at least 20 errors at the target error ratio, so short patterns report "Insufficient data" together with the number of symbols needed (PRBS20 and longer are usually enough); electrical probes only; licensing and metering are the same as for FEC analysis (Chapter 4).

When comparing with COM, mind the receiver structure: with OP.MLSE on, COM 4.x deploys its own RxFFE solution whose first residual post-cursor is the trellis coefficient b1 (the internal blim(1); the CSV column DFE_taps reads 0). REA adopts the same receiver by default when importing such a CSV (import dialog "COM receiver for MLSE cases" / SCPI CONF:COM:IMPORT:MLSERx) and derives that residual as a 1-tap DFE, which the MLSE block then uses as b1. The measured gain is usually well below the CSV delta-COM: the 178A model feeds the residual ISI into the error-event union bound as a stationary noise process with a power spectrum, whereas on the real waveform the residual ISI is deterministic and already removed by ideal decision feedback, so the detector can only gain on the genuine noise components. That is exactly the difference between the analytic and the measured number;

both are kept for comparison.

Appendix A PLTS Alignment Overview

This appendix summarizes REA alignment with Keysight PLTS 2024 U1. All S-parameter processing defaults match PLTS Default Settings, providing PLTS-equivalent behavior out of the box.

A.1 PLTS-aligned Defaults

Module	Setting	REA default = PLTS default
TDR	Window type	Nominal (= PLTS Exponential, Coef 0.50)
TDR	Window mode	Lowpass (step)
TDR	DC extrapolation	Linear (= PLTS Auto)
TDR	Causality enforcement	Off by default (PLTS does not enforce)
TDR	Velocity factor	0.70
TDR	Display	Impedance (= PLTS Step Impedance)
Cascade/Deembed/Split	4-port order	Adjacent pairing P1,P3 -> P2,P4 (= PLTS Adjacent)
Gating	Gate mode	Bandpass (= PLTS Bandpass)
Smith	Chart type	Impedance
Smith	Reference impedance	50 ohm SE (auto 100 ohm for diff)
Smith	VSWR circles	On (1.5/2/3/5)

A.2 TDR Output Formats (PLTS plot equivalents)

REA mode	PLTS equivalent	Description
impedance	STEP_IMPEDANCE_NS	Impedance Z (Ω), default
voltage	STEP_VOLTS_NS	Voltage V (V■=0.4V)
step	STEP_REAL_NS	Step reflection coefficient Γ
logmag	STEP_LOGMAG_NS	Reflection log |Γ| (dB)

REA mode	PLTS equivalent	Description
impulse	IMPULSE_REAL_NS	Impulse response
impulse_logmag	IMPULSE_LOGMAG_NS	Impulse log |h| (dB)

A.3 PLTS-equivalent Feature Map

REA Feature	PLTS Equivalent
Time-domain gating on the TDR page (see 9.3)	Tools -> Time-Domain Gating
"Reverse subsequent networks" option on the Cascade page	Embedding "Reverse" checkbox
Smith chart admittance mode	Inverted Smith (admittance)
Smith hover marker readout (Z/Gamma/VSWR/dB/RL)	Marker readout
Smith VSWR circles + automatic reference impedance	draw_vswr + system Z0 config

A.4 Numerical Consistency

REA has been verified against PLTS 2024 U1 scenario by scenario: cascade and de-embedding agree at machine precision; time-domain gating and TDR impedance curves match point-by-point in typical scenarios (isolated multi-section transition structures show ohm-level differences, see A.5); Smith chart readouts are byte-identical.

A.5 Remaining Differences

- **TDR 1-3 ohm deviation at transitions**: at multi-section impedance transitions the two tools use different time-domain reconstruction approaches; a small known difference remains.
- **Multi-section asymmetric ratio split**: the splitting algorithm must treat a multi-section structure as an "equivalent uniform line" - an information-theoretic limit common to all such tools.
- **PLTS short segment TDR rejection**: extreme ratios (r=0.1/0.9) produce segments too short electrically; PLTS internally skips computation.

Appendix B Glossary

Plain-language definitions of the recurring terms in this manual, written for readers new to signal-integrity (SI) analysis.

Term	Plain-language meaning
S-parameters	Network data describing how much of a signal passes through / reflects off a channel at each frequency; measured with a VNA or exported from an EM simulator
Touchstone (.s2p/.s4p)	The standard S-parameter file format; the digit is the port count
Insertion / return loss	Attenuation through the channel / portion reflected back (dB)
Differential signal / Sdd	Data carried on a pair of opposite-phase wires; Sdd is the equivalent 2-port S-parameter of the differential mode
Port pairing	Telling the software which two ports of a 4-port file form one differential pair
NRZ / PAM4	Two-level (1 bit/symbol) / four-level (2 bits/symbol) modulation
Baud rate (Gbaud)	Symbols per second; PAM4 bit rate = 2 x baud rate
UI (Unit Interval)	The time occupied by one symbol; the x-axis unit of an eye diagram
Eye diagram	All symbol periods of a long bit stream overlaid on one plot - the wider the "eye" opens, the better the signal
Eye height / width	Vertical / horizontal size of the trace-free region in the eye center
ISI (inter-symbol interference)	Insufficient channel bandwidth smears symbols into their neighbors - the main reason high-speed eyes close
PRBS	Pseudo-random bit sequence (e.g. PRBS7 = length 2^7-1), used to emulate real traffic
Jitter (RJ / SJ)	Random / sinusoidal displacement of signal edges in time
SSC	Spread-spectrum clocking - deliberately sweeping the clock to reduce EMI peaks

Term	Plain-language meaning
FFE	Feed-forward equalizer: weights and sums neighboring symbols to pre-cancel channel smearing (TX or RX side)
pre/post-cursor	FFE taps before / after the main symbol
CTLE	Continuous-time linear equalizer: an analog filter boosting highs and attenuating lows
DFE	Decision feedback equalizer: subtracts the residual tail of already-decided symbols
CDR	Clock data recovery: the receiver loop that extracts the sampling clock from data
COM	Channel Operating Margin (dB), defined by IEEE 802.3 - the core pass/fail metric for a channel
VEO	Vertical eye opening at the target BER (mV)
SNDR	Signal to noise-plus-distortion ratio (dB)
FEXT / NEXT	Far-/near-end crosstalk: neighboring channels coupling into the victim
ICN	Integrated crosstalk noise - crosstalk folded into one equivalent RMS noise number
BER	Bit error ratio: fraction of bits received wrong (e.g. 1e-12)
TDECQ	802.3 eye-closure metric for PAM4 optical transmitters (dB), measured by a reference receiver
ER (extinction ratio)	Optical power ratio of the "1" level to the "0" level (dB)
OMA	Optical modulation amplitude: power difference between optical "1" and "0"
RIN	Laser relative intensity noise (dB/Hz)
TDR	Time-domain reflectometry: impedance-vs-distance profile computed from S-parameters