

Simulation Improvement Playbook

A reusable checklist for making any simulation correct, explorable, and worth keeping

Scientific Inquiry with AI

Additional Support File — V1

Created by LLNL Summer 2026 STEM Education Research Team

Student Team: Ramina Amino, Jahanvi Chamria, Arya Ferozy, Bryanna Gonzalez, Tai Le, Zedikiah McAdams, John Navarra, Joshua Sarabia, Abdurrahman Raza

Faculty Team: Praveen Pathak, David Rakestraw, David Strubbe, Brian Utter

How to use this. Once your simulation runs and passes its validation checks, walk this list to make it genuinely useful. It's organized by category. Each suggestion is a Do paired with the Don't it replaces, tagged by priority, with a copy-paste prompt to make the change. You'll reuse this on every simulation you build, whatever the topic, science or not, not just this one.

Tier	Meaning
MUST	Do these or the simulation isn't trustworthy or usable. Non-negotiable.
SHOULD	Do these to make it a real instrument for inquiry, not just a demo.
NICE	Polish that pays off when you reuse or share it.

1. Scientific Validity & Trust

Tier	Do / Don't	Prompt to get there
MUST	Do: Use the full governing model. Don't: Let the AI silently substitute a simplifying approximation (e.g. small-angle, ideal-gas, linearization) without telling you.	Use the full governing rules for [phenomenon]. Do not apply any simplifying approximation unless I ask. State in a comment any assumption you made.
MUST	Do: Show a live numeric readout of the quantity your validation depends on. Don't: Hide the result behind animation, so you can only eyeball it.	Display a live numeric readout of [period / total energy / conserved sum / key metric], updating as the simulation runs.
MUST	Do: Make a known/limiting case reachable from the controls. Don't: Bake in settings where you can't recover a case with a known answer.	Make sure I can set [the parameters] to the case where the answer is known to be [value], so I can check it.
SHOULD	Do: Surface a conserved quantity or fixed total so drift is visible. Don't: Assume conservation holds without ever displaying it.	Compute and plot [the conserved quantity / fixed total] over time so I can see whether it stays constant.

2. Controls & Interactivity

Tier	Do / Don't	Prompt to get there
MUST	Do: Expose the real parameters as sliders or inputs, with units on every label. Don't: Hard-code magic numbers a user can't change.	Expose [parameters] as labeled sliders with units and sensible ranges. Let me also type exact values.
MUST	Do: Add a reset button that returns to a known starting state. Don't: Force a page reload to start over.	Add a Reset button that restores the initial state and clears any plots.
SHOULD	Do: Let inputs reach the extremes where the model should fail. Don't: Clamp ranges to only the well-behaved regime.	Extend the slider ranges so I can push [parameter] to extreme values where the model may break.
NICE	Do: Add play / pause / step and a speed control. Don't: Lock the animation to one speed you can't follow.	Add play, pause, single-step, and a speed control for the animation.

3. Visual Clarity & Readout

Tier	Do / Don't	Prompt to get there
MUST	Do: Label axes, units, and what each visual element represents. Don't: Leave an unlabeled curve or blob the viewer has to guess at.	Label all axes with quantities and units, and add a short legend for every plotted element.
SHOULD	Do: Plot the output over time or over the swept parameter, not just an animation. Don't: Rely on animation alone, which is hard to measure.	Add a graph of [output] versus [time / swept parameter] alongside the animation.
NICE	Do: Mark the predicted value or known case on the plot. Don't: Make the user compare against a number in their head.	Draw a reference line on the plot at the predicted value [value] so I can compare at a glance.

4. Reusability Across Topics

Tier	Do / Don't	Prompt to get there
MUST	Do: Keep it one self-contained HTML file you download, not just an in-app Artifact/Canvas preview. Don't: Leave it as a live view inside the AI app, or split it across files or require a build step or API key.	Keep everything in a single self-contained HTML file with inline JS and CSS, no external libraries, runs offline. Give me the full file to download.
SHOULD	Do: Print the assumptions and the domain of validity on screen. Don't: Leave the model's limits implicit so future-you forgets them.	Add an on-screen note listing the key assumptions and the range where this model is valid.
SHOULD	Do: Add a one-line header: what this models and where it breaks. Don't: Hand yourself an unlabeled black box you won't understand next month.	Add a one-line title and subtitle stating what this simulates and where it stops being accurate.
NICE	Do: Provide preset scenario buttons for common cases. Don't: Make every demo a manual slider hunt.	Add preset buttons that load [a few named scenarios] with one click.

Why “downloaded HTML file,” not “Artifact / Canvas / Gem”? An Artifact (Claude) or Canvas (ChatGPT, Gemini) is the live preview beside the chat, handy for building, but it lives in that AI account, may need a login, and can change or disappear. (Gemini Gems and custom GPTs are saved chatbot personalities, not a simulation file.) A single downloaded .html file runs offline anywhere, looks the same for everyone, and is yours to keep, re-open, and submit. Build in the live view if you like, then have the AI output the whole thing as one self-contained HTML file and download it.

Companion to this lesson: Damped_Pendulum_Simulation.html is the reference simulation for “Building to Understand: Creating AI-Generated Simulations as a Mode of Scientific Inquiry,” and Pendulum_Simulation_Prompt_and_Build_Log.docx shows every Do below applied against a real build, with its outcome recorded.

5. Robustness & Reliability

Tier	Do / Don't	Prompt to get there
MUST	Do: Handle bad or extreme input gracefully (no crashes, no blank screen). Don't: Let a divide-by-zero or NaN silently freeze the simulation.	Guard against invalid or extreme inputs: prevent NaN/Infinity, and show a clear message instead of crashing.
SHOULD	Do: Use a stable, small-enough time step (or an adaptive method) so results don't depend on it. Don't: Pick a step size that makes the result drift or blow up.	Use a numerically stable integration step (or an adaptive method); confirm the result doesn't change if I halve the step.
NICE	Do: Show the current numeric error or step size when relevant. Don't: Hide numerical settings that affect accuracy.	Display the integration step size and let me change it, so I can see its effect on accuracy.

6. Pedagogical Usefulness (Learning)

Tier	Do / Don't	Prompt to get there
SHOULD	Do: Let the user predict, then reveal, e.g. a 'show answer' or target-matching mode. Don't: Give away the result before the user commits to a guess.	Add a mode where I set a prediction first, then reveal the actual result and the difference.
SHOULD	Do: Highlight the non-obvious behavior (the threshold, resonance, saturation, tipping point) when it appears. Don't: Bury the interesting regime among ordinary ones.	Visually flag when [the interesting behavior] occurs as I sweep [parameter].
NICE	Do: Let the user export their data or a snapshot for a report. Don't: Trap the results inside the page.	Add a button to export the current data as CSV and to save a snapshot image of the plot.

Remember the order: validate first, improve second. None of this polish matters if the underlying model is wrong, make it correct, then make it useful.

One-Glance Checklist

Before you call a simulation done, confirm:

- **MUST** — full model (no hidden approximations) · live readout of your check quantity · a reachable known/limiting case · labeled controls with units · reset · graceful on bad input · one self-contained HTML file you can download.
- **SHOULD** — a conserved quantity or fixed total shown · a plot vs. time or swept parameter · inputs reach the failure regime · assumptions & validity on screen · stable time step · predict-then-reveal.
- **NICE** — play/pause/step · reference line at the predicted value · preset scenarios · data/image export.