DPA26TZ05-DV003
SPEED DIAL
Frequently Asked Questions (FAQs)

Q1: For the Small Business Technology Transfer team, how should responsibilities be divided between the research institution's discovery-engine research and the small business's production platform, workflow integration, and commercialization activities?
A1: Proposers must define the division of responsibilities while strictly adhering to standard STTR allocation requirements (e.g., minimum 40% small business, minimum 30% research institution).

Q2: What verification, reproducibility, version-control, and benchmarking infrastructure is required before an automatically discovered algorithm can enter a trusted engineering workflow?
A2: Proposers must identify, propose, and justify the necessary infrastructure required to ensure trust, verification, and successful transition into operational engineering workflows.
Q3: Which engineering environments must be supported first (MATLAB/Simulink, COMSOL, Python, C++, other), and what level of auto-code generation/compilation/deployment is expected?
A3: Target environments and the extent of auto-compilation are dependent on the proposer's selected domain. Proposers must justify their choices based on their proposed domains.

Q4: What is the desired relationship between the discovered-algorithm library and online/on-demand discovery: retrieve-and-adapt, warm-start new searches, or fully rediscover per engineering problem?
A4: There is no mandated relationship. Proposers should define a viable concept of operations that optimally supports their target domains and workflow integration.

Q5: How should physical constraints, engineering requirements, hardware constraints, and certification requirements be encoded into the algorithm-discovery search space?
A5: Encoding methods are domain-dependent. Proposers must define and justify their approach to integrating physical, engineering, and hardware constraints into the discovery process.

Q6: What discovery-engine paradigms must the platform support - transformers, reinforcement learning, genetic programming, program synthesis, differentiable programming, or an extensible plug-in architecture?
A6: No specific paradigms are recommended. Proposers are responsible for proposing and justifying an architecture that best meets the topic objectives.

Q7: How will DARPA evaluate human interpretability of discovered algorithms, and what representation or proof/verification artifacts are expected?
A7: Interpretability will be evaluated based on the algorithm's composition of transparent, mathematically provable building blocks that domain experts can understand, verify, and certify.

Q8: What quantitative performance evidence is sufficient to establish Phase-I-equivalent feasibility, and must the prior discovered algorithms be in the same application domains proposed for Phase II?

A8: Proposers must provide written evidence (e.g., technical reports, peer-reviewed publications, or independent technical reviews) of their capability to discover novel algorithms. Prior discovered algorithms are not required to be in the same application domain proposed for Phase II, provided the proposer can substantiate that the capability translates effectively.

Q9: What constitutes a “novel algorithm” for Direct-to-Phase-II feasibility - a new symbolic/mathematical structure, a learned algorithmic policy, an automatically composed solver, or any method that demonstrably outperforms known baselines?
A9: A novel algorithm is a new method that systematically solves a family of problems, supported by rigorous mathematical backing. Methods based on simple heuristics, rigid conditional logic (e.g., standard if/then statements), or "black-box" approaches without a mathematical explanation are not of interest.

Q10: What classes of scientific/engineering algorithms should Phase II prioritize for discovery and deployment (optimization, control, estimation, signal processing, partial differential equation solvers, linear algebra, or other domains)?
A10: No specific classes of algorithms are prioritized. Proposers must identify the classes of algorithms relevant to their selected application domain and justify their dual-use potential.