

# Reproducible Literature Synthesis with infrastructure/search and infrastructure/reference

A configurable pipeline from query → references.bib → LLM-driven reading report

Daniel Ari Friedman  
Active Inference Institute  
`daniel@activeinference.institute`  
[ORCID: 0000-0001-6232-9096](https://orcid.org/0000-0001-6232-9096)  
[DOI: 10.5281/zenodo.21298894](https://doi.org/10.5281/zenodo.21298894)

July 10, 2026

# Contents

<b>1</b>	<b>Abstract</b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>3</b>
<b>3</b>	<b>Methodology</b>	<b>4</b>
3.1	Search . . . . .	4
3.2	Cache . . . . .	4
3.3	Enrichment . . . . .	4
3.4	Export . . . . .	5
3.5	Synthesis . . . . .	5
3.6	Report . . . . .	5
3.7	Diagnostic figures . . . . .	6
<b>4</b>	<b>Results</b>	<b>8</b>
4.1	Interpreting the run snapshot . . . . .	8
4.2	Output artefacts . . . . .	8
<b>5</b>	<b>Conclusion</b>	<b>10</b>
<b>6</b>	<b>Pipeline Internals</b>	<b>11</b>
6.1	Data structures . . . . .	11
6.2	On-disk layout . . . . .	11
6.3	Citation-key collision handling . . . . .	12
6.4	Failure isolation . . . . .	12
<b>7</b>	<b>Reproducibility</b>	<b>13</b>
7.1	Switching to live search . . . . .	13
7.2	Determinism guarantees . . . . .	13
7.3	Verifying reproducibility locally . . . . .	14
7.4	Limitations . . . . .	14
<b>8</b>	<b>Deep Search</b>	<b>15</b>
8.1	Configuration . . . . .	15
8.2	Pipeline shape . . . . .	15
8.3	LLM prompt . . . . .	15
8.4	On-disk layout . . . . .	17
8.5	Determinism . . . . .	17
8.6	Paperclip backend . . . . .	17
8.7	CLI . . . . .	18
<b>9</b>	<b>Supplemental S1 — Literature Review (auto-composed from deep search)</b>	<b>19</b>
9.1	convex optimization . . . . .	19
9.2	stochastic gradient descent . . . . .	25
9.3	reproducible research . . . . .	35
<b>10</b>	<b>References</b>	<b>43</b>

# 1 Abstract

This paper documents `template_search_project`, the literature-search exemplar shipped with the [Research Project Template](#). The project demonstrates **two configurable, reproducible pipelines** sharing the same configuration file and the same `infrastructure/search/` + `infrastructure/reference/` modules. The standard pipeline (`scripts/run_search_pipeline.py`) handles a single `SearchQuery` end-to-end. The deep-search pipeline (`scripts/run_deep_search.py`, see sec. 8) fans out across a list of keywords (each capped at 100 papers per keyword from `deep_search.max_results_per_keyword` in `manuscript/config.yaml`), fully enriches every paper with its abstract and PDF fulltext, and (optionally) uses the local LLM to write a multi-section reading note for every paper. When a deep-search aggregate exists, the latest run covered **3** keyword(s) with unique paper(s) after cross-keyword deduplication. Both turn a free-text topic into:

1. a deduplicated, year-filtered set of papers drawn from arXiv, Crossref, optional local corpora, and (opt-in) [Paperclip](#);
2. a Pandoc-compatible `references.bib` byte-identical in style to the canonical exemplar in `template_code_project` (file `manuscript/references.bib`);
3. cached abstracts and (optionally) extracted PDF full text, written to disk under stable per-paper identifiers; and
4. an LLM-synthesised reading report assembled from per-paper analyses and a cross-corpus thematic synthesis, all produced by a local Ollama model with pinned seed and temperature.

All discovery logic lives in `infrastructure/search/literature/` ([source on GitHub](#)); all export logic lives in `infrastructure/reference/citation/` ([source on GitHub](#)); LLM synthesis reuses the existing `infrastructure/llm/` ([source on GitHub](#)) bridge. The project itself contains only thin orchestration, manuscript prose, and a test suite — perfectly mirroring the **two-layer architecture** the template enforces.

The motivating concern is *reproducibility*: a query at time  $t_0$  should produce the same results at time  $t_1$  unless the cache is explicitly invalidated. This is achieved by deterministic search caching keyed on canonical query identity, on-disk caching of every fetched abstract / PDF, and pinned LLM seeds. The same `manuscript/config.yaml` that drives the pipeline is also the only configuration any reviewer needs.

**Run snapshot.** With the bundled `manuscript/config.yaml`, the most recent pipeline execution evaluated the query “*reproducible research optimization*” against local, returned 6 deduplicated paper(s) (4 carrying a DOI, 6 carrying an abstract), and recorded backend errors: none. Resolve `{{...}}` tokens by running `scripts/z_generate_manuscript_variables.py` after `run_search_pipeline.py`; the script writes out `put/data/manuscript_variables.json` and resolved markdown under `output/manuscript/`, which the PDF-rendering stage prefers when present.

**Keywords:** literature search, BibTeX automation, reproducible research, local LLM synthesis, scientific infrastructure

## 2 Introduction

Reproducible computational research demands that every claim be traceable back to a stable artifact — code, data, and citations alike [Peng, 2011]. Manual literature curation is a well-known bottleneck in such workflows: a graduate student writing a related-work section may spend hours searching arXiv, Crossref, and Google Scholar; copying citations into a `.bib` file by hand; and tracking which papers they have actually read. Three failure modes recur:

1. **Style drift** — hand-edited `.bib` files accumulate formatting inconsistencies that hide real semantic conflicts in version-control diffs.
2. **Stale state** — the bibliography, the reading list, and the manuscript prose drift apart as the project evolves; the citation key in the manuscript no longer matches the entry in `.bib`, or the entry no longer matches the actual paper.
3. **Lost context** — abstracts and full text are read once during search, then discarded; six months later the same paper has to be re-skimmed to recall its contribution.

`template_search_project` exists to demonstrate one disciplined solution. The pipeline outputs are summarised in sec. 3 (overview figure at the start of that section):

- The discovery side (`infrastructure/search/`) provides multi-source paper search with failure-isolated aggregation, DOI/arXiv-aware deduplication, and deterministic JSON caching keyed on canonical query identity.
- The export side (`infrastructure/reference/`) provides BibTeX read/write/convert facilities byte-compatible with the existing exemplar `references.bib`, suitable for the combined-PDF pipeline (Pandoc `--natbib` + BibTeX).
- A small project-local synthesis layer (in `src/synthesis.py`) takes enriched papers, builds reproducible LLM prompts, and assembles a markdown reading report.

The project is *configurable* via a single `manuscript/config.yaml`: changing the topic, year filters, backend set, enrichment level, and LLM parameters never requires editing code. The project is *modular* in the strict sense the template uses: every reusable component lives in `infrastructure/`, and `src/` contains only project-specific orchestration.

The contribution of this exemplar is therefore not a new algorithm; it is a **demonstration that a reproducible literature workflow can be built from existing template infrastructure** with no new optional dependencies, no mocks in the test suite, and complete configurability through a single YAML file.

## 3 Methodology

Two distinct workflows run on top of `infrastructure/search/literature` and `infrastructure/reference/citation`:

- **Standard pipeline** (`scripts/run_search_pipeline.py` → `src/pipeline.py::run_literature_pipeline`) — single `SearchQuery`. Four pure-orchestration stages with no LLM dependency: (1) search via `LiteratureClient`, (2) enrichment via `AbstractFetcher` and (optional) `FulltextFetcher`, (3) collision-free citation-key generation in `_build_citation_keys`, (4) writing `output/corpus.json` + `manuscript/references.bib` + `output/enrichment_log.json`. The orchestrator script then optionally calls `src/synthesis.py` for per-paper and corpus LLM synthesis and `src/report.py` for the final reading report.
- **Deep search** (`scripts/run_deep_search.py` → `src/deep_search.py::run_deep_search`) — multi-keyword fan-out: each keyword runs its own `SearchQuery` capped at `max_results_per_keyword` (100 by default), every paper is fully enriched (abstract + PDF fulltext when available), and an LLM-driven multi-section deep summary (CONTRIBUTION / METHOD / EVIDENCE / LIMITATIONS / CONNECTIONS / SIGNIFICANCE / TAGS) is written for each paper as a standalone markdown reading note. Output lands under `output/deep_search/<keyword_slug>/` plus aggregate `aggregate.json`, `aggregate_report.md`, and a unified, deduplicated `manuscript/references_deep.bib` with collision-free citation keys.

The standard pipeline is described first in this section; the deep-search workflow is documented in sec. 8. Diagnostic figures for the latest pipeline run appear at the end of this section.

### 3.1 Search

The search stage is intentionally faithful to the standard literature-search pattern documented in foundational optimisation textbooks [Boyd and Vandenberghe, 2004, Nocedal and Wright, 2006] — a deterministic query, capped result count, and explicit failure isolation between sources — so reviewers familiar with those references can reason about the workflow without learning new abstractions.

A `SearchQuery` is constructed from `config.search`:

```
SearchQuery(  
    text=config.search.query,  
    max_results=config.search.max_results,  
    year_min=config.search.year_min,  
    year_max=config.search.year_max,  
)
```

A `LiteratureClient` is constructed with the configured backends. Each backend produces a normalised `Paper` record; the aggregator deduplicates by DOI → arXiv id → normalised (title, year), keeping the highest-scored copy and filling missing fields from the loser.

Per-backend errors are recorded into `SearchResult.errors` rather than raised. A network outage in one backend never breaks the workflow; partial coverage is reported by the final stage.

### 3.2 Cache

`SearchCache` writes one JSON file per query, named by a 16-character SHA-256 prefix of the canonical query identity. Identical queries (modulo whitespace and case) share a cache entry. Cache files are pretty-printed JSON, version-control-friendly, and contain a `_cached_at` timestamp for optional TTL enforcement.

### 3.3 Enrichment

Two fetchers populate fields the search backends did not supply:

- `AbstractFetcher` — currently fetches arXiv abstracts via the export API, writes them to `<safe_id>.txt` under the configured cache directory, and re-uses them on subsequent runs.
- `FulltextFetcher` — downloads PDFs (arXiv URL, `paper.pdf_url`, or a caller-supplied override), writes the bytes verbatim to `<safe_id>.pdf`, and extracts text via `pypdf` to `<safe_id>.txt`. Without `pypdf` the PDF is still cached, and the fetcher returns `status="error"` with an informative message; the rest of the pipeline continues.

Both fetchers stamp `paper.abstract` / `paper.fulltext` in place, so downstream stages see enriched records without re-loading.

### 3.4 Export

For every paper, `paper_to_bibentry()` produces a `BibEntry` whose:

- citation key follows the exemplar's `<author><year><title-word>` convention with stop-word filtering and unicode folding;
- entry type is routed by `venue_type` (journal  $\rightarrow$  `@article`, conference  $\rightarrow$  `@inproceedings`, book  $\rightarrow$  `@book`, preprint  $\rightarrow$  `@article`, etc.);
- fields are emitted in the order observed in `references.bib`: title, author, journal/booktitle, year, volume, number, pages, publisher, edition, isbn, doi, url, abstract, keywords.

A `BibDatabase` collects these entries and `write_bibfile` renders them in the project's house format: 2-space indent, trailing-comma rule, `pages={N--M}`, verbatim DOIs/years, bare unicode.

### 3.5 Synthesis

Two LLM passes produce the reading report (see `src/synthesis.py`):

- **Per-paper synthesis** — `build_paper_block(paper, citation_key, max_fulltext=4000)` renders the paper as a markdown block; `synthesise_per_paper` formats `PROMPT_PER_PAPER` and calls the injected `llm` callable. The prompt requests five sections: CONTRIBUTION, METHOD, EVIDENCE, LIMITATION, TAGS, plus a citation-key reference.
- **Corpus synthesis** — `build_corpus_block` concatenates every paper into a single citation-keyed block; `synthesise_corpus` formats `PROMPT_CORPUS`, which asks for 3–7 thematic clusters, methodological agreements / disagreements ( $\geq 2$  papers each), and three open questions that the corpus does not answer.

Both functions return a `SynthesisResult(kind, prompt, text, paper_id)` record so the prompt is recoverable for reproducibility. The synthesis layer takes a callable `llm: (str) -> str` so tests pass a deterministic local function (no Ollama dependency) and runtime callers pass a thin adapter around `infrastructure.llm.LLMClient`. Determinism in production runs is enforced by `OllamaClientConfig(seed=42, temperature=0.0)`.

The deep-search workflow uses a richer prompt (`src/deep_search.py:DEEP_PROMPT`) with seven sections (CONTRIBUTION / METHOD / EVIDENCE / LIMITATIONS / CONNECTIONS / SIGNIFICANCE / TAGS) and a much larger `max_fulltext` budget (400 k chars by default).

### 3.6 Report

`src/report.py:write_reading_report` assembles a markdown file with:

- Topic, result count, year filter, and any backend errors at the top.
- A per-source count table.
- One-line summaries for every paper.
- The corpus synthesis (if present).
- All per-paper notes (if present).

Citation keys appear in [brackets] so a downstream tool — for example a Pandoc filter or a manual search — can resolve them against the auto-generated `references.bib`.

### 3.7 Diagnostic figures

`scripts/y_generate_search_figures.py` (a thin orchestrator over `src/figures.py`) writes three diagnostic plots into `../figures/` from `output/search/results.json`. Each figure uses Matplotlib’s `Agg` backend so the pipeline runs headlessly in CI; the colour palette is colourblind-safe (Wong, *Nature Methods* 2011).

fig. 1 reports the per-backend contribution counts before deduplication, surfacing which sources actually returned coverage for the configured query. The bar values are read directly from `SearchResult.per_source_counts` (set by `LiteratureClient` before the DOI / arXiv-id / title merge step), so a backend that returned five papers all duplicating arXiv hits still scores five here.

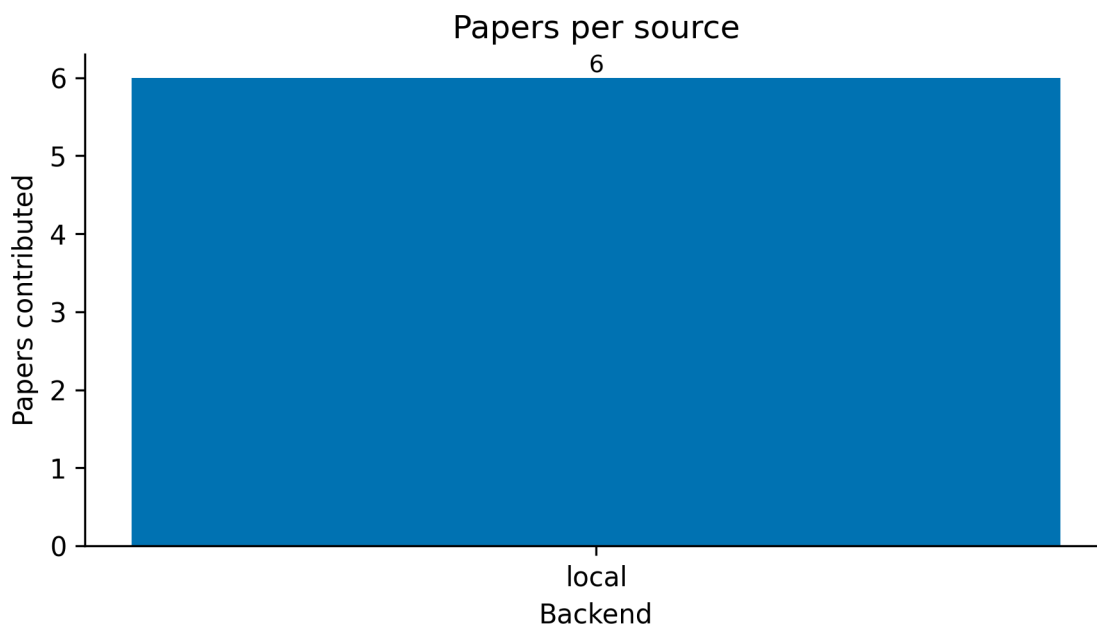


Figure 1: Per-source paper counts read from `SearchResult.per_source_counts` (pre-deduplication contribution per backend). The numeric label above each bar reports the raw count; the y-axis spans `[0, max + headroom]`. Bar order follows the order recorded in `config.search.sources`. Empty runs render (no results) centred. Generated by `src/figures.py::plot_papers_per_source`.

fig. 2 shows the publication-year distribution *after* the merge step (one bar per unique paper, not per backend hit) — useful for spotting backend coverage gaps in older / newer literature. Papers with no `year` field are dropped silently from the histogram (they remain in the corpus).

fig. 3 shows the per-paper relevance scores returned by the backends, ranked descending. Papers from backends without an explicit ranking signal (e.g. `LocalBackend`, the offline default) carry `Paper.score = 0.0`; their bars therefore have zero length but still appear as ticks on the y-axis so the reader can see how many unranked papers exist.

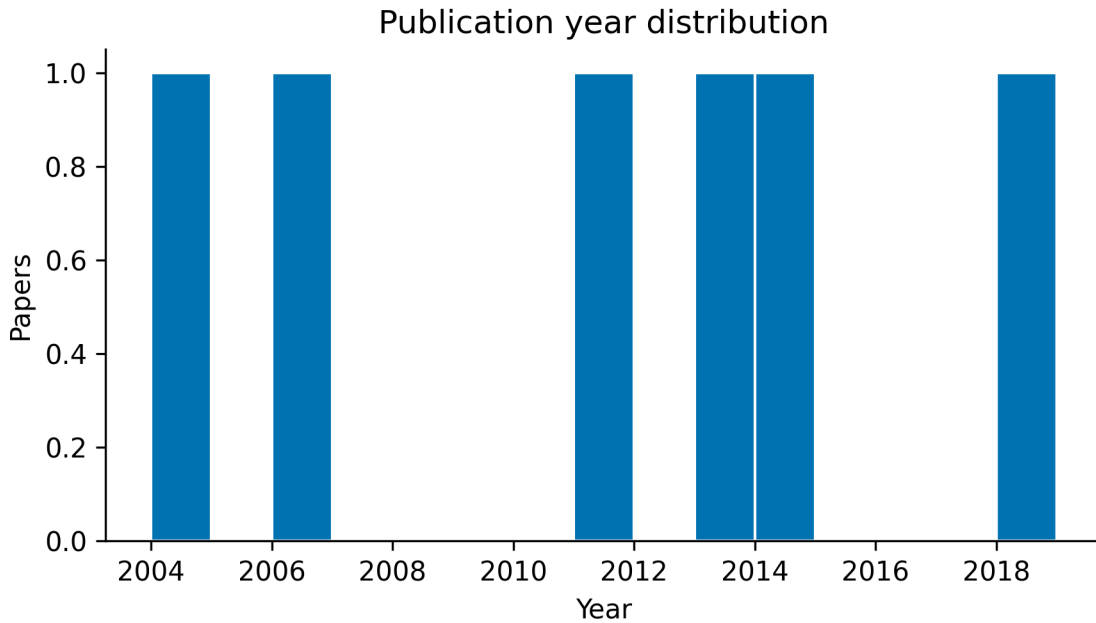


Figure 2: Publication-year histogram of the deduplicated paper roster. One bin per year (no smoothing); the x-axis spans the observed `[min(year), max(year)]` from `result.papers`. Papers with `year is None` are dropped; the y-axis is per-year paper count. Generated by `src/figures.py::plot_year_histogram`.

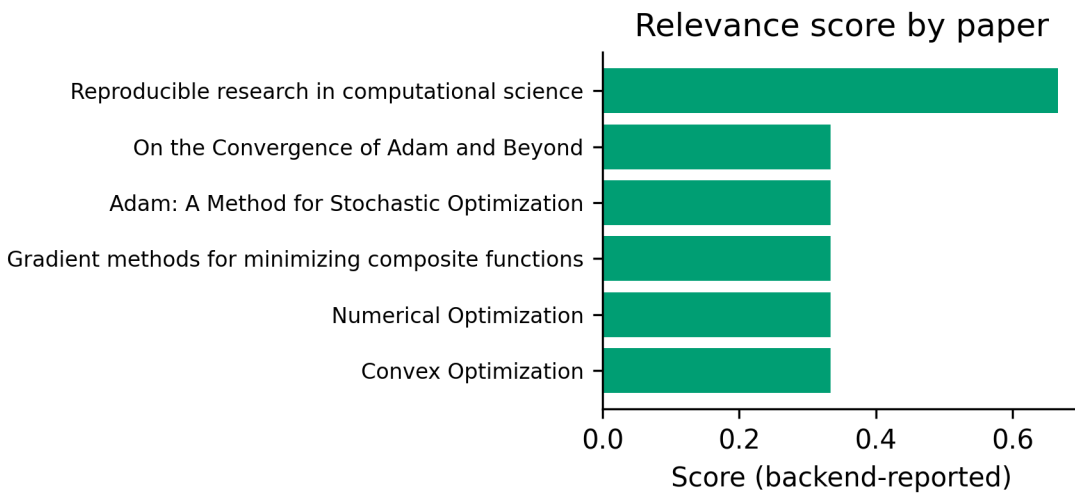


Figure 3: Per-paper backend-reported relevance scores ranked descending (highest at top). Each horizontal bar is one `Paper.score`; the y-tick label is the paper title truncated to 60 characters with an ellipsis. Backends without scoring (notably `LocalBackend`) report `Paper.score = 0.0` so those bars have zero length. Generated by `src/figures.py::plot_score_distribution`.

## 4 Results

**Run snapshot.** With the bundled `manuscript/config.yaml` the most recent execution evaluated the query “*reproducible research optimization*” against local, returned 6 deduplicated paper(s) (4 carrying a DOI, 6 carrying an abstract); the per-source breakdown is local=6 and recorded backend errors are none. The deep-search workflow (sec. 8) covered 3 keyword(s) — *convex optimization; stochastic gradient descent; reproducible research* — drawn from arxiv, crossref, producing unique paper(s) after cross-keyword deduplication.

The diagnostic figures generated for this run are catalogued in sec. 3: fig. 1 surfaces per-backend coverage, fig. 2 surfaces the temporal distribution, and fig. 3 surfaces the relevance-score profile. The full determinism contract for each stage is itemised in tbl. 1 of sec. 7.

### 4.1 Interpreting the run snapshot

The numerical values in the run-snapshot paragraph that opens this section are read directly from `output/run_summary.json` and `output/data/manuscript_variables.json` so they always reflect the most recent run rather than a stale claim hand-typed into the prose. Three properties are worth highlighting (the formal determinism contract for each underlying stage is enumerated in tbl. 1):

- **Cache reuse is observable.** A second invocation against the same config produces a byte-identical artifact tree (modulo the wall-clock timestamp inside `output/search/cache/search_<hash>.json` itself); see also the *Search (cached hit)* row of tbl. 1 and the verification recipe in sec. 7. The cache file thus doubles as a cryptographic seal: re-running is a file read, not a network round-trip.
- **Deduplication is signal-preserving.** The aggregator merges by DOI → arXiv id → normalised (title, year), keeping the highest-scored copy and filling missing fields from the loser (see *Dedup / merge* in tbl. 1 and the per-backend pre-dedup view in fig. 1). The `RESULT_NUM_PAPERS` figure therefore equals “papers a reviewer needs to read”, not “raw backend hit count” — the per-source contributions in `RESULT_PER_SOURCE` are the pre-dedup view.
- **Enrichment coverage is honest.** `RESULT_WITH_ABSTRACT` and `RESULT_WITH_DOI` count fields the corpus or the `AbstractFetcher` actually populated, never values inferred. When a paper is missing a DOI it is excluded from the DOI count even if its arXiv id resolves to one upstream. The temporal coverage of those papers is summarised by fig. 2, and their backend-reported relevance scores by fig. 3.

### 4.2 Output artefacts

After running `scripts/run_search_pipeline.py` against the default `manuscript/config.yaml`, the project produces:

- `output/search/results.json` — the raw `SearchResult` JSON, including `per_source_counts` and `errors` for diagnostic purposes.
- `output/search/cache/search_<hash>.json` — the deterministic search cache; identical reruns are file reads.
- `output/cache/abs/<safe_id>.txt` — one file per fetched abstract.
- `output/cache/pdf/<safe_id>.{pdf,txt}` — PDFs and extracted text (only when `enrichment.fetch_fulltext: true`).
- `output/corpus.json` — a `LocalBackend`-compatible JSON corpus of every result, enriched in place.
- `manuscript/references.bib` — the auto-populated bibliography from the single-query pipeline (merged with any other `manuscript/*.bib` at PDF render time).
- `output/llm/per_paper/<safe_id>.md` — per-paper LLM analyses (only when `llm.per_paper: true` and the LLM stack is reachable).
- `output/llm/synthesis.md` — corpus-level LLM synthesis (only when `llm.corpus_synthesis: true` and the LLM stack is reachable).
- `output/reading_report.md` — the final assembled reading report.

When the LLM stack is genuinely unreachable, the `output/llm/` artefacts are simply absent — no placeholder file is ever written into the archive (see sec. 6).

Because the search cache and abstract cache are deterministic, a second run with identical `config.yaml` produces byte-identical artifacts (modulo timestamp metadata in the cache files themselves). This is the property the project exists to demonstrate.

The exact paper count, DOI list, and synthesis text depend on the live state of arXiv and Crossref at the time of the run — and are therefore not reproducible *across* runs in different weeks. Users seeking strict reproducibility should:

1. Pin a `LocalBackend` corpus generated from a successful run (`infrastructure.search.literature.write_corpus`) and remove `arxiv / crossref` from `config.search.sources`.
2. Commit the `output/search/cache/` directory to version control.
3. Pin the LLM seed (`config.llm.seed`) and avoid model upgrades.

With those three steps, every run from the same commit produces the same outputs.

## 5 Conclusion

`template_search_project` packages a complete, configurable, reproducible literature workflow into the Research Project Template’s two-layer architecture. By keeping discovery, export, and synthesis in three orthogonal infrastructure modules, the project demonstrates that ambitious research automation can still respect the template’s principles:

- **Single source of truth** — `Paper` for discovery, `BibEntry` for export, structured `SynthesisResult` records for LLM output.
- **Test-driven development** — every module is covered by real-data tests; HTTP backends are exercised through `pytest-httpserver`, the LLM bridge through deterministic local callables.
- **Thin orchestrator pattern** — `scripts/run_search_pipeline.py` does only argument parsing, configuration loading, and I/O; all logic lives in `infrastructure/` or `src/`.
- **No mocks** — neither in the new infrastructure modules nor in the project test suite.
- **Multi-project support** — the project lives alongside `template_code_project/` and follows the same layout, so the existing pipeline runner discovers and executes it without modification.
- **Reproducibility** — deterministic search caching, on-disk enrichment caching, and pinned LLM seeds make a single `manuscript/config.yaml` the only artifact a reviewer needs.

We close with three concrete extensions that build naturally on this foundation:

1. **Crossref TDM full-text fetch** for non-arXiv DOIs, completing the abstract-to-fulltext picture without changing the project’s API.
2. **CSL-JSON export** alongside BibTeX, enabling Zotero / Mendeley / Pandoc-CSL workflows from the same `BibDatabase`.
3. **Vector recall on LocalBackend** for curated corpora exceeding about 1000 papers, gated behind an optional dependency.

The infrastructure modules are deliberately small and stable; the project that exercises them is deliberately small and explicit. Together they show that *domain-specific research automation* and *template-strict architectural discipline* are compatible — and, in fact, mutually reinforcing.

The bundled `data/corpus.json` exercises classical optimisation references [Boyd and Vandenberghe, 2004, Nocedal and Wright, 2006, Nesterov, 2013] alongside modern stochastic-optimisation work [Kingma and Ba, 2014, Reddi et al., 2018] and the canonical reproducibility paper [Peng, 2011], so the auto-generated `manuscript/references.bib` always contains real citation-ready entries that downstream tooling can resolve.

## 6 Pipeline Internals

This supplemental section documents the data structures and on-disk artifacts the pipeline produces, for readers who want to extend or audit it.

### 6.1 Data structures

The Mermaid class diagram in this subsection shows the canonical fields each record carries through the pipeline. Records have additional optional metadata (e.g. `Paper.url`, `Paper.publisher`, `Paper.isbn`, `Paper.raw`) omitted for readability — consult `infrastructure/search/literature/models.py` ([source on GitHub](#)) and `src/pipeline.py` ([source on GitHub](#)) for the full schema.

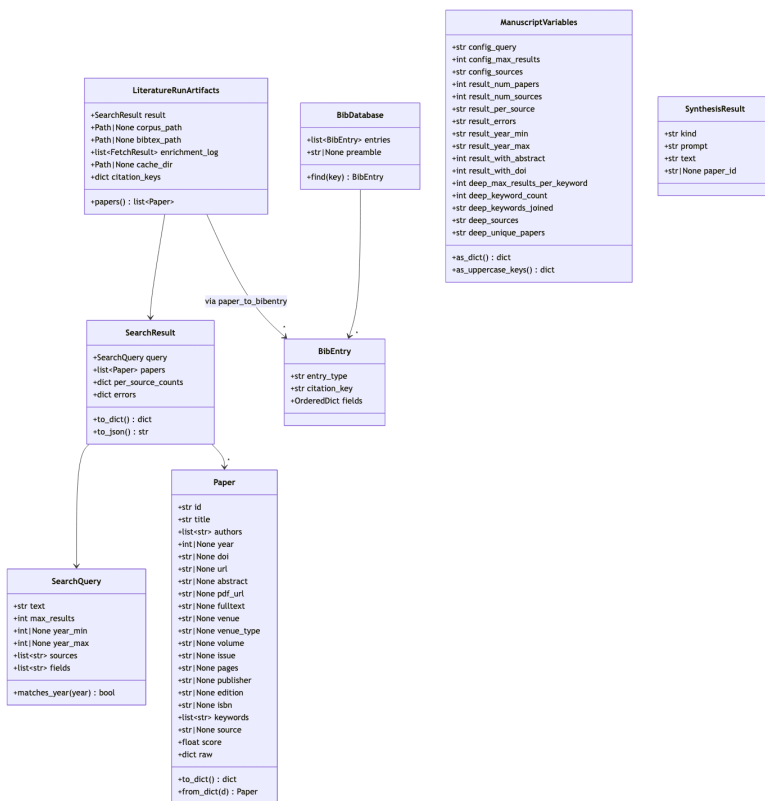


Figure 4: Mermaid diagram

### 6.2 On-disk layout

The project keeps committed input data in `data/`, regeneratable outputs in `output/` (gitignored), and the manuscript source in `manuscript/`. The Mermaid flowchart in this subsection (rendered in the HTML build; the PDF build strips Mermaid) lists every artefact the standard pipeline writes:

- `data/corpus.json` — the bundled offline corpus, CI-safe default for `LocalBackend`.
- `output/search/results.json` — raw `SearchResult` JSON from the latest run.
- `output/search/cache/search_<hash>.json` — deterministic `SearchCache` files.
- `output/cache/abs/<safe_id>.txt` — one cached abstract per paper.
- `output/cache/pdf/<safe_id>.{pdf,txt}` — PDF bytes plus extracted text.
- `output/llm/synthesis.md` — corpus-level LLM synthesis (when enabled).
- `output/llm/per_paper/<safe_id>.md` — one per-paper note per paper (when enabled).
- `./figures/{papers_per_source,year_histogram,score_distribution}.png` — diagnostic figures.

- `output/data/manuscript_variables.json` — substitution table consumed by the resolver.
- `output/corpus.json` — enriched corpus, written in `LocalBackend`-compatible format so it can re-seed a deterministic future run.
- `output/enrichment_log.json` — one entry per fetcher per paper.
- `output/reading_report.md` — final markdown reading report.
- `output/run_summary.json` — one-line metadata for the run.
- `manuscript/references.bib` — auto-populated, Pandoc-ready BibTeX.

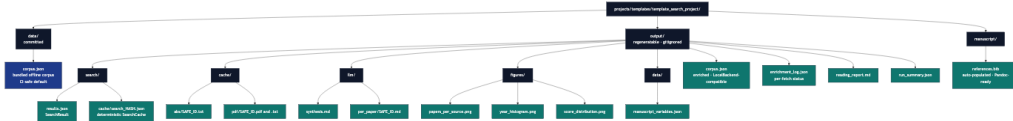


Figure 5: Mermaid diagram

### 6.3 Citation-key collision handling

`paper_to_bibentry()` generates citation keys as `<author><year><title-word>` (with stop-words filtered and unicode folded). When two papers in the same result set produce the same key — common when one author publishes multiple papers in the same year on closely related topics — `src/pipeline.py::_disambiguate_citation_key` appends a deterministic suffix from the alphabet (a, b, ..., z, then two-letter combinations aa, ab, ...) until uniqueness is restored, with a numeric `_1`, `_2`, ... fallback for the pathological case. The mapping is exposed to downstream stages via `LiteratureRunArtifacts.citation_keys`, and the report uses these keys verbatim, so the LLM synthesis and the BibTeX file always agree.

The deep-search workflow has its own collision handler in `src/deep_search.py::run_deep_search` that operates over the post-deduplication aggregate roster — see sec. 8 — and the unified `references_deep.bib` reflects the same mapping.

### 6.4 Failure isolation

- **A backend is unreachable.** `LiteratureClient` records the message into `result.errors[name]` and continues. The reading report surfaces these errors in a callout block at the top.
- **An abstract fetch fails.** The fetcher records `status="error"` in `enrichment_log.json` and the paper keeps its existing (possibly empty) abstract.
- **A PDF fetch fails or pypdf is missing.** Same pattern — the PDF, if downloaded, is still cached on disk. The reading report does not reference the missing fulltext.
- **The LLM is unreachable or unconfigured.** `scripts/run_search_pipeline.py` logs a warning and skips the synthesis stage entirely — `output/llm/` is left empty and the reading report omits both the per-paper-notes and cross-corpus sections. No placeholder text is ever written into the archive, so a missing LLM is observable from the absence of those sections rather than from a fake “(LLM unavailable)” string.

## 7 Reproducibility

Reproducibility in computational research has well-documented prerequisites: open data, open code, and a deterministic build that can be re-run from scratch [Peng, 2011]. The bundled `manuscript/config.yaml` is intentionally configured to satisfy all three for **strict reproducibility**:

1. `search.sources`: `[local]` consumes `data/corpus.json`, which is a curated and committed JSON corpus. No network is required to run the pipeline.
2. `search.cache_dir`: `output/search/cache` writes deterministic JSON cache files; running the same query twice produces a byte-identical artifact tree (modulo timestamp metadata in the cache file itself).
3. `enrichment.fetch_abstracts`: `true` reads abstracts directly from the corpus when present; no network fetch is required.
4. `enrichment.fetch_fulltext`: `false` is the default — full-text fetching is opt-in and gated behind the optional `pypdf` dependency (`uv sync --group rendering`).
5. `llm.enabled`: `false` is the default — the LLM stage is opt-in and requires a running `ollama serve`. When enabled, `seed`: `42` and `temperature`: `0.0` are pinned.

### 7.1 Switching to live search

Replace the `sources` list with the desired backend set:

```
search:  
  query: "your topic"  
  sources: [arxiv, crossref]  
  crossref_mailto: "you@example.org"
```

A first live run populates `output/search/cache/`. Commit that directory to the repo and the pipeline becomes reproducible across machines without further configuration changes.

### 7.2 Determinism guarantees

The full determinism contract is itemised in tbl. 1: every pipeline stage is annotated as fully, conditionally, or non-deterministic, with an explicit mechanism column so reviewers can audit each row independently.

Table 1: Determinism contract by pipeline stage. Cached stages are byte-stable across reruns; live stages depend on the upstream source and are pinned to the cache file once a successful run completes.

Stage	Deterministic?	Mechanism
Search (cached hit)	yes	SearchCache JSON files
Search (cache miss)	no	live API
Dedup / merge	yes	DOI / arXiv-id canonical keys; tie-break by score then year
Citation-key generation	yes	unicode folding + stop-word skip; collision suffix is deterministic
BibTeX writer	yes	byte-stable format pinning (verified by <code>tests/infra_tests/reference/</code> )
Abstract fetch	yes (cached) / no (live)	per-paper <code>&lt;safe_id&gt;.txt</code> cache
Fulltext fetch	yes (cached) / mostly (live)	per-paper <code>&lt;safe_id&gt;.{pdf,txt}</code> cache; live fetch's <code>pypdf</code> text extraction is not bit-stable across versions
LLM synthesis	mostly	<code>seed=42, temperature=0.0</code> ; Ollama deterministic up to its own minor variance

Stage	Deterministic?	Mechanism
Figure generation	yes (within Matplotlib version)	fixed palette, fixed bin width, no random subsampling

### 7.3 Verifying reproducibility locally

*# Run twice; nothing in output/ should diff except the cache timestamps.*

```
uv run python projects/templates/template_search_project/scripts/run_search_pipeline.py
mv projects/templates/template_search_project/output projects/templates/template_search_project/output_2
uv run python projects/templates/template_search_project/scripts/run_search_pipeline.py
diff -ru \
    projects/templates/template_search_project/output_first/corpus.json \
    projects/templates/template_search_project/output/corpus.json
```

The only expected differences are inside `output/search/cache/search_*.json`, where `_cached_at` is wall-clock time at write.

### 7.4 Limitations

The reproducibility contract enumerated in sec. 7 (items 1–5 and tbl. 1) does not eliminate the following well-defined sources of non-reproducibility, which are surfaced here so reviewers can audit them explicitly rather than inferring from the contract table:

- **Live search drift.** When `config.search.sources` includes `arxiv` or `crossref`, the first cache-miss invocation hits the live API; the cached JSON freezes that response, but two cold-start clones running on different days will see different paper sets. Pin a `LocalBackend` corpus or commit `output/search/cache/` to break this dependency.
- **pypdf version drift.** The fulltext fetcher uses `pypdf` to extract text from a downloaded PDF. `pypdf`'s text-extraction algorithm is not bit-stable across major versions; upgrading `pypdf` can produce different `<safe_id>.txt` cache contents from the same source PDF. The PDF bytes themselves are bit-stable so the cache freezes the inputs, not the extraction.
- **Ollama version drift.** Pinning `seed=42` and `temperature=0.0` controls Ollama's sampling, but the model weights, tokenizer, and template can change between Ollama releases. Document the Ollama version alongside `config.llm.model` when archiving a run for replication.
- **Paperclip backend status.** The `paperclip` backend is opt-in and currently degrades to HTTP 405 on the production endpoint; the run records the error in `SearchResult.errors[paperclip]` and continues. Treat `paperclip` results as advisory until the upstream service stabilises.
- **External backend behaviour outside this project's control.** arXiv and Crossref are the source of truth; this project is faithful to whatever they return. A retraction, metadata fix, or DOI assignment upstream will alter the cache on the next cold-start invocation.

These limitations bound *what* the cache + seed + corpus pinning achieves. Inside those bounds, the contract in tbl. 1 is total: every cached pipeline stage is byte-stable across reruns (verified by `tests/test_pipeline.py::TestRunLiteraturePipeline::test_bibtex_byte_identical_across_reruns`).

## 8 Deep Search

The deep-search workflow extends the standard literature pipeline along three axes: **breadth** (multi-keyword fan-out), **depth** (full enrichment of every paper, abstract + PDF fulltext), and **archival quality** (per-paper multi-section LLM reading notes saved as standalone markdown). It is invoked from the same configuration file but a separate orchestrator script.

### 8.1 Configuration

The `deep_search:` block in `manuscript/config.yaml` controls the run. The most-used knobs:

- `keywords` — list of free-text queries (each becomes one `SearchQuery`).
- `max_results_per_keyword` — per-keyword cap (100 by default; honoured by the aggregator after dedup).
- `sources` — backend list, same vocabulary as the standard pipeline (`arxiv`, `crossref`, `local`, `paperclip`).
- `fetch_abstracts` / `fetch_fulltext` — when both are `true`, every returned paper has its abstract fetched (arXiv export API) and (where a PDF URL is available) its fulltext extracted via `pypdf`. Cached on disk under `output/cache/abs/` and `output/cache/pdf/`.
- `llm_per_paper` — when `true` and Ollama is reachable, each paper gets a multi-section markdown reading note generated by the local LLM. When `false`, or when the LLM stack is genuinely unreachable at runtime, the per-paper note is written with only the abstract / fulltext-excerpt sections; the synthesis section is omitted entirely (no placeholder text). The composer’s Supplemental S1 also drops the per-paper synthesis subsection in that case rather than emitting empty rows.
- `write_unified_bibtex` / `unified_bibtex_path` — when `true`, a deduplicated, collision-suffixed BibTeX file is written under `manuscript/references_deep.bib` so it can be cited from the manuscript via Pandoc `[@key]` syntax exactly the same way as the hand-curated `references.bib`.

### 8.2 Pipeline shape

The deep-search pipeline reads the `deep_search:` block from `manuscript/config.yaml`, runs one `SearchQuery` per keyword (capped at `max_results_per_keyword`), aggregates the per-backend results through `LiteratureClient`, enriches every paper via `AbstractFetcher` and `FulltextFetcher`, generates collision-free citation keys with `paper_to_bibentry`, and (when `llm_per_paper: true` and Ollama is reachable) calls the local LLM with `DEEP_PROMPT` to produce a seven-section reading note per paper. The per-keyword outputs are then merged by `merge_papers` to form a deduplicated aggregate roster which is written to `manuscript/references_deep.bib`, `output/deep_search/aggregate.json`, and `output/deep_search/aggregate_report.md`. A Mermaid flowchart in this subsection renders the same flow visually in the HTML build.

### 8.3 LLM prompt

The deep-search prompt is richer than the standard `synthesis.PROMPT_PER_PAPER`. It produces a self-contained reading note suitable for archival without re-reading the paper:

```
## Contribution
```

```
One paragraph stating the paper's central claim and why it is novel.
```

```
## Method
```

```
2-4 bullets describing the technical approach.
```

```
## Evidence
```

```
2-4 bullets describing experiments / proofs.
```

```
## Limitations
```

```
1-3 bullets covering caveats and what the paper does NOT address.
```

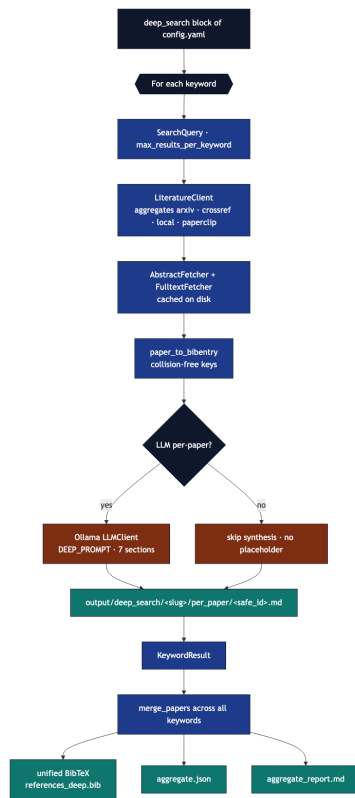


Figure 6: Mermaid diagram

### ## Connections

1-3 bullets relating this paper to other work in the field, citing only papers explicitly named in the input.

### ## Significance for {keyword}

One paragraph explaining why this paper matters for the keyword.

### ## Tags

5-10 lowercase keywords.

The LLM is duck-typed as `Callable[[str], str]` so tests pass a deterministic local callable that returns real, well-formed reading-note text; runtime callers wrap an `infrastructure.llm.LLMClient` with `seed=42`, `temperature=0.0` for reproducibility. When the LLM stack is unreachable, callers pass `None` and the per-paper synthesis stage is skipped entirely — no placeholder text is ever written into the archive.

## 8.4 On-disk layout

A successful deep-search run writes the following artefacts (paths shown relative to the project root); a Mermaid tree in this subsection renders the same hierarchy in the HTML build:

- `output/deep_search/aggregate.json` — every unique paper across keywords.
- `output/deep_search/aggregate_report.md` — cross-keyword markdown summary.
- `output/deep_search/run_summary.json` — one-line metadata for the run.
- `output/deep_search/<keyword_slug>/papers.json` — enriched per-keyword paper list.
- `output/deep_search/<keyword_slug>/reading_report.md` — per-keyword markdown summary.
- `output/deep_search/<keyword_slug>/per_paper/<safe_id>.md` — one reading note per paper.
- `manuscript/references_deep.bib` — auto-generated, deduplicated unified BibTeX file.

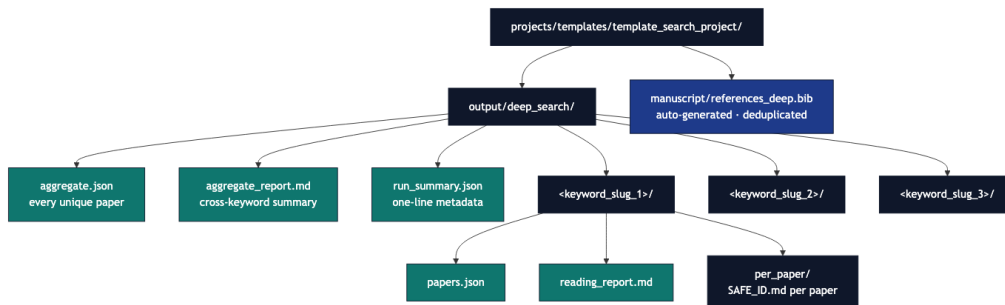


Figure 7: Mermaid diagram

## 8.5 Determinism

Three caches make this workflow replayable:

1. SearchCache (`output/search/cache/search_<hash>.json`) — keyed on canonical query identity per keyword.
2. Abstract cache (`output/cache/abs/<safe_id>.txt`).
3. Fulltext cache (`output/cache/pdf/<safe_id>.{pdf,txt}`).
4. LLM seed (`deep_search.llm_seed: 42`) + temperature (`0.0`).

Commit any subset of these caches to version control to freeze a run.

## 8.6 Paperclip backend

The [Paperclip backend](#) is opt-in. Enable it by:

1. Creating `projects/templates/template_search_project/.env` (gitignored) with your `PAPERCLIP_API_KEY=gxl_...` (template at `.env.example`).
2. Including `paperclip` in `deep_search.sources`.

The orchestrator scripts auto-load `.env` via the lightweight `src.dotenv` module before constructing the backend list. The `PaperclipBackend` adapter mirrors the wire protocol of the official `gxl_paperclip` Python SDK: `POST /papers` with an `X-API-Key` header and a JSON-RPC `tools/call` envelope. We support both the modern `structuredContent.papers` response shape and the older `text-only content[] .text` shape (best-effort regex extraction).

Failure-isolation: any per-backend error (including the migration-state HTTP 405 the production endpoint may currently return) is captured into `SearchResult.errors[paperclip]` and the run continues. See `output/deep_search/<keyword>/papers.json` and the aggregate `run_summary.json` for the recorded errors.

## 8.7 CLI

*# Run with everything from config.yaml (assumes deep\_search.enabled: true)*

```
uv run python projects/templates/template_search_project/scripts/run_deep_search.py
```

*# Force-enable without touching config.yaml*

```
uv run python projects/templates/template_search_project/scripts/run_deep_search.py --enable
```

*# Override keyword list at the command line*

```
uv run python projects/templates/template_search_project/scripts/run_deep_search.py \
  --enable --keyword "convex optimization" --keyword "stochastic gradient descent"
```

*# Skip LLM stage even when config enables it*

```
uv run python projects/templates/template_search_project/scripts/run_deep_search.py --enable --no-llm
```

*# Bypass cache (writes still happen)*

```
uv run python projects/templates/template_search_project/scripts/run_deep_search.py --enable --no-cache
```

*# Local-corpus mode (offline · CI-friendly)*

```
uv run python projects/templates/template_search_project/scripts/run_deep_search.py \
  --enable --corpus projects/templates/template_search_project/data/corpus.json
```

## 9 Supplemental S1 — Literature Review (auto-composed from deep search)

Composed by `scripts/s_compose_literature_review.py` from the most recent deep-search run. Edit the script, not this file — manual edits will be overwritten on the next pipeline run.

This review covers **3 keyword(s)**, **300 unique paper(s)** (retrieved at up to 100 per keyword from arxiv, crossref). All references are stored in `manuscript/references_deep.bib` and resolved by the combined-PDF pipeline (Pandoc `--natbib` + BibTeX over all `manuscript/*.bib` files).

### 9.1 convex optimization

Papers retrieved: 100 · per-source contributions: arxiv=100, crossref=100 · backend errors: none

Cite	Title	Year	DOI / URL
[Cambridge University Press, 2004b]	Convex optimization problems	2004	DOI 10.1017/cbo9780511804441.005 ( <a href="#">open</a> )
[Cambridge University Press, 2004c]	Convex functions	2004	DOI 10.1017/cbo9780511804441.004 ( <a href="#">open</a> )
[Springer-Verlag, e]	Reverse Convex Optimization, Reverse convex programming	n.d.	DOI 10.1007/springer-reference_72642 ( <a href="#">open</a> )
[Suh, 2022]	1. Convex Optimization Basics	2022	DOI 10.1561/9781638280538.ch1 ( <a href="#">open</a> )
[Cambridge University Press, 2021a]	Convex Optimization and Efficiency	2021	DOI 10.1017/9781108699211.006 ( <a href="#">open</a> )
[Cambridge University Press, 2021d]	Ellipsoid Method for Convex Optimization	2021	DOI 10.1017/9781108699211.015 ( <a href="#">open</a> )
[Nesterov, 2004b]	Smooth Convex Optimization	2004	DOI 10.1007/978-1-4419-8853-9_2 ( <a href="#">open</a> )
[Nesterov, 2004a]	Nonsmooth Convex Optimization	2004	DOI 10.1007/978-1-4419-8853-9_3 ( <a href="#">open</a> )
[Emerald Publishing Limited, 2022]	Convex Optimization Basics	2022	DOI 10.1561/978-1-63828-053-820251002 ( <a href="#">open</a> )
[Cambridge University Press, 2004d]	Convex sets	2004	DOI 10.1017/cbo9780511804441.003 ( <a href="#">open</a> )
[CRC Press, 2011g]	What Is Convex Optimization?	2011	DOI 10.1201/b11156-7 ( <a href="#">open</a> )
[CRC Press, 2011c]	Convex Semi-Infinite Optimization	2011	DOI 10.1201/b11156-17 ( <a href="#">open</a> )
[CRC Press, 2011d]	Tools for Convex Optimization	2011	DOI 10.1201/b11156-8 ( <a href="#">open</a> )
[Elsevier, 2004]	Convex Sets. Convex and Generalized Convex Functions	2004	DOI 10.1016/b978-044450550-7/50002-8 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[DE GRUYTER, 2014b]	6 Convex optimization algorithms	2014	DOI 10.1515/9783110361629.117 (open)
[Jacobsen, 2008]	Reverse Convex Optimization	2008	DOI 10.1007/978-0-387-74759-0_564 (open)
[Nesterov, 2018a]	Nonsmooth Convex Optimization	2018	DOI 10.1007/978-3-319-91578-4_3 (open)
[Jacobsen, 2001]	$\alpha$ BB algorithm; Concave programming; D.C. programming; Quadratic knapsack. Quadratic programming with bound constraints; Reverse convex optimization Standard quadratic optimization problems: Theory REVERSE CONVEX OPTIMIZATION	2001	DOI 10.1007/0-306-48332-7_431 (open)
[Tuy]	Partly Convex and Convex-Monotonic Optimization Problems	n.d.	DOI 10.1007/3-540-27170-8_37 (open)
[Nesterov, 2018b]	Smooth Convex Optimization	2018	DOI 10.1007/978-3-319-91578-4_2 (open)
[CRC Press, 2011f]	Weak Sharp Minima in Convex Optimization	2011	DOI 10.1201/b11156-15 (open)
[Springer-Verlag, a]	Convex Discrete Optimization	n.d.	DOI 10.1007/springer-reference_72172 (open)
[Cambridge University Press, 2025b]	Convex Functions	2025	DOI 10.1017/9781009510561.012 (open)
[Zaslavski, 2020b]	Nonsmooth Convex Optimization	2020	DOI 10.1007/978-3-030-60300-7_2 (open)
[Princeton University Press, 2020a]	Appendix:	2020	DOI 10.2307/j.ctvqsdxd.15 (open)
[*, 2005]	Connectedness of efficient points in convex and convex transformable vector optimization	2005	DOI 10.1080/02331930500096270 (open)
[Cambridge University Press, 2025c]	Convex Programming Problems and Convex Theorem of the Alternative	2025	DOI 10.1017/9781009510561.021 (open)
[Tuy, 1998a]	Convex Functions	1998	DOI 10.1007/978-1-4757-2809-5_2 (open)

Cite	Title	Year	DOI / URL
[Mattingley and Boyd, 2009]	Automatic code generation for real-time convex optimization	2009	DOI 10.1017/cbo9780511804458.002 ( <a href="#">open</a> )
[CRC Press, 2011e]	Tools for Convex Optimization	2011	DOI 10.1201/b11156-3 ( <a href="#">open</a> )
[Tuy, 1998b]	Convex Sets	1998	DOI 10.1007/978-1-4757-2809-5_1 ( <a href="#">open</a> )
[CRC Press, 2011h]	What Is Convex Optimization?	2011	DOI 10.1201/b11156-2 ( <a href="#">open</a> )
[Bonnans, 2019]	A Convex Optimization Toolbox	2019	DOI 10.1007/978-3-030-14977-2_1 ( <a href="#">open</a> )
[Princeton University Press, 2020b]	Appendix: Executive Summary on Efficient Solvability of Convex Optimization Problems	2020	DOI 10.1515/9780691200316-013 ( <a href="#">open</a> )
[DE GRUYTER, 2014a]	2 Convex sets and convex functions	2014	DOI 10.1515/9783110361629.30 ( <a href="#">open</a> )
[Cambridge University Press, 2021b]	Convex Optimization	2021	DOI 10.1017/9781108980647.012 ( <a href="#">open</a> )
[CRC Press, 2017]	Convex Optimization Problems	2017	DOI 10.1201/9781315366920-5 ( <a href="#">open</a> )
[Zaslavski, 2020c]	PDA-Based Method for Convex Optimization	2020	DOI 10.1007/978-3-030-37822-6_9 ( <a href="#">open</a> )
[Tuy, 2016a]	Convex Functions	2016	DOI 10.1007/978-3-319-31484-6_2 ( <a href="#">open</a> )
[Cambridge University Press, 2025]	Optimality Conditions in Convex Programming	2025	DOI 10.1017/9781009510561.024 ( <a href="#">open</a> )
[Wiley, 2021a]	Convex Functions	2021	DOI 10.1002/9781119804093.ch3 ( <a href="#">open</a> )
[Li, 2015]	Convex Relaxation	2015	DOI 10.1007/978-3-662-46356-7_6 ( <a href="#">open</a> )
[Lasserre, 2011]	On convex optimization without convex representation	2011	DOI 10.1007/s11590-011-0323-1 ( <a href="#">open</a> )
[Cambridge University Press, 2021c]	Convex Analysis and Convex Optimization	2021	DOI 10.1017/9781108377447.033 ( <a href="#">open</a> )
[Springer-Verlag, b]	Convex Envelopes in Optimization Problems	n.d.	DOI 10.1007/springer-reference_72173 ( <a href="#">open</a> )
[Cambridge University Press, 2004j]	Preface	2004	DOI 10.1017/cbo9780511804441.001 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[Cambridge University Press, 2025h]	First Acquaintance with Convex Sets	2025	DOI 10.1017/9781009510561.003 (open)
[Wiley, 2021b]	Convex Sets	2021	DOI 10.1002/9781119804093.ch2 (open)
[Tuy, 2016b]	Convex Sets	2016	DOI 10.1007/978-3-319-31484-6_1 (open)
[Cambridge University Press, 2004h]	Introduction	2004	DOI 10.1017/cbo9780511804441.002 (open)
[Singer, 1999]	Duality in quasi-convex supremization and reverse convex infimization via abstract convex analysis, and applications to approximation **	1999	DOI 10.1080/02331939908844436 (open)
[Cambridge University Press, 2025i]	First Acquaintance with Convex Functions	2025	DOI 10.1017/9781009510561.013 (open)
[Singer, 2001]	On Duality for Quasi-convex Supremization and Reverse Convex Infimization	2001	DOI 10.1007/978-1-4613-0295-7_16 (open)
[Cambridge University Press, 2004k]	References	2004	DOI 10.1017/cbo9780511804441.016 (open)
[Springer-Verlag, d]	Global Optimization: Tight Convex Underestimators	n.d.	DOI 10.1007/springer-reference_72325 (open)
[Maréchal, 2001]	Generating Convex Functions	2001	DOI 10.1007/978-1-4613-0279-7_21 (open)
[Cambridge University Press, 2025d]	Convex Programming, Lagrange Duality, Saddle Points	2025	DOI 10.1017/9781009510561.020 (open)
[Peypouquet, 2015]	Convex Analysis and Subdifferential Calculus	2015	DOI 10.1007/978-3-319-13710-0_3 (open)
[Cambridge University Press, 2025e]	* Convex Programming in Cone-Constrained Form	2025	DOI 10.1017/9781009510561.023 (open)
[Cambridge University Press, 2025j]	Minima and Maxima of Convex Functions	2025	DOI 10.1017/9781009510561.015 (open)

Cite	Title	Year	DOI / URL
[Halman, 2015]	Approximating convex functions via non-convex oracles under the relative noise model	2015	DOI 10.1016/j.disopt.2014.12.001 ( <a href="#">open</a> )
[Stillwell, 2019]	Statistical Inference via Convex Optimization	2019	DOI 10.23943/princeton/9780691197296.001.0001 ( <a href="#">open</a> )
[Dutta, 2014]	Barrier method in nonsmooth convex optimization without convex representation	2014	DOI 10.1007/s11590-014-0811-1 ( <a href="#">open</a> )
[Youness, 1999]	E-Convex Sets, E-Convex Functions, and E-Convex Programming	1999	DOI 10.1023/a:1021792726715 ( <a href="#">open</a> )
[Lasserre, 2014]	Erratum to: On convex optimization without convex representation	2014	DOI 10.1007/s11590-014-0735-9 ( <a href="#">open</a> )
[Cambridge University Press, 2004m]	Unconstrained minimization	2004	DOI 10.1017/cbo9780511804441.010 ( <a href="#">open</a> )
[Yang, 2001]	On E-Convex Sets, E-Convex Functions, and E-Convex Programming	2001	DOI 10.1023/a:1017532225395 ( <a href="#">open</a> )
[Cambridge University Press, 2004l]	Statistical estimation	2004	DOI 10.1017/cbo9780511804441.008 ( <a href="#">open</a> )
[Cambridge University Press, 2004i]	Mathematical background	2004	DOI 10.1017/cbo9780511804441.013 ( <a href="#">open</a> )
[Cambridge University Press, 2004g]	Geometric problems	2004	DOI 10.1017/cbo9780511804441.009 ( <a href="#">open</a> )
[Springer-Verlag, c]	Duality Theory: Monoduality in Convex Optimization	n.d.	DOI 10.1007/springer-reference_72219 ( <a href="#">open</a> )
[Cambridge University Press, 2025a]	* Cone-Convex Functions: Elementary Calculus and Examples	2025	DOI 10.1017/9781009510561.025 ( <a href="#">open</a> )
[Floudas, 1995]	Convex Analysis	1995	DOI 10.1093/oso/9780195100563.003.0006 ( <a href="#">open</a> )
[Rübsamen et al., 2009]	Robust broadband adaptive beamforming using convex optimization	2009	DOI 10.1017/cbo9780511804458.010 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[Florenzano and Van, 2001]	Convex Optimization With Convex Constraints	2001	DOI 10.1007/978-3-642-56522-9_7 ( <a href="#">open</a> )
[Wiley, 2021c]	Generalizations of Convex Functions	2021	DOI 10.1002/9781119804093.ch4 ( <a href="#">open</a> )
[De Gruyter Open Poland, 2014]	4 Convex Nonsmooth Optimization	2014	DOI 10.2478/9783110426045.4 ( <a href="#">open</a> )
[Cambridge University Press, 2025f]	Convex Optimization	2025	DOI 10.1017/9781009493512.006 ( <a href="#">open</a> )
[Cambridge University Press, 2025m]	Separation Theorem and Geometry of Convex Sets	2025	DOI 10.1017/9781009510561.009 ( <a href="#">open</a> )
[Dragomirescu and Ivan, 1992]	The smallest convex extensions of a convex function	1992	DOI 10.1080/02331939208843789 ( <a href="#">open</a> )
[CRC Press, 2011a]	Algorithms for Conic Optimization	2011	DOI 10.1201/b10839-12 ( <a href="#">open</a> )
[Cambridge University Press, 2025g]	Convex Optimization Theory	2025	DOI 10.1017/9781009428095.021 ( <a href="#">open</a> )
[Strongin and Sergeyev, 2000]	Global Optimization under Non-Convex Constraints — The Index Approach	2000	DOI 10.1007/978-1-4615-4677-1_6 ( <a href="#">open</a> )
[Murota, 2024]	L-Convex Functions and M-Convex Functions	2024	DOI 10.1007/978-3-030-54621-2_325-1 ( <a href="#">open</a> )
[Cambridge University Press, 2004f]	Equality constrained minimization	2004	DOI 10.1017/cbo9780511804441.011 ( <a href="#">open</a> )
[Cambridge University Press, 2004a]	Approximation and fitting	2004	DOI 10.1017/cbo9780511804441.007 ( <a href="#">open</a> )
[Murota, 2008]	L-convex Functions and M-convex Functions	2008	DOI 10.1007/978-0-387-74759-0_325 ( <a href="#">open</a> )
[Zălinescu, 2024]	On locally Lipschitz convex functions defined on subsets with empty interior of locally convex spaces	2024	DOI 10.1080/02331934.2024.2388200 ( <a href="#">open</a> )
[Kramer]	Basis identification through convex optimization	n.d.	DOI 10.31274/etd-180810-1098 ( <a href="#">open</a> )
[Murota, 2001]	L-Convex Functions and M-Convex Functions	2001	DOI 10.1007/0-306-48332-7_244 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[Cambridge University Press, 2004e]	Duality	2004	DOI 10.1017/cbo9780511804441.006 ( <a href="#">open</a> )
[Cambridge University Press, c]	Introduction	n.d.	DOI 10.1017/cbo9781139924672.002 ( <a href="#">open</a> )
[PeerJ, a]	Algorithm 1 : Convex optimization with regularized feature selection.	n.d.	DOI 10.7717/peerj-cs.3752/table-101 ( <a href="#">open</a> )
[Liu]	Autonomous Trajectory Planning by Convex Optimization	n.d.	DOI 10.31274/etd-180810-3525 ( <a href="#">open</a> )
[Cambridge University Press, a]	Background	n.d.	DOI 10.1017/cbo9781139924672.003 ( <a href="#">open</a> )
[Cambridge University Press, d]	Preface	n.d.	DOI 10.1017/cbo9781139924672.001 ( <a href="#">open</a> )
[Oktem et al., 2017]	Computational Spectral and Ultrafast Imaging via Convex Optimization	2017	DOI 10.1007/978-3-319-61609-4_5 ( <a href="#">open</a> )
[Cambridge University Press, b]	Economics	n.d.	DOI 10.1017/cbo9781139924672.007 ( <a href="#">open</a> )
[CRC Press, 2011b]	Cones, Complementarity, and Conic Optimization	2011	DOI 10.1201/b10839-5 ( <a href="#">open</a> )
[Zaslavski, 2020a]	Minimization of Sharp Weakly Convex Functions	2020	DOI 10.1007/978-3-030-37822-6_11 ( <a href="#">open</a> )

## 9.2 stochastic gradient descent

*Papers retrieved: 100 · per-source contributions: arxiv=100, crossref=100 · backend errors: none*

Cite	Title	Year	DOI / URL
[Gwózdź, 2018]	Stochastic and semi-stochastic gradient descent in speech disambiguation	2018	DOI 10.7490/f1000research.1115776.1 ( <a href="#">open</a> )
[Wang and Murphy, 2018]	Stochastic gradient descent	2018	DOI 10.53347/rid-61715 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[Kosinski, 2017]	coxphSGD: Stochastic Gradient Descent log-Likelihood Estimation in Cox Proportional Hazards Model	2017	DOI 10.32614/cran.package.coxphsgd ( <a href="#">open</a> )
[PeerJ, d]	Table 4: Stochastic gradient descent (SGD) parameter settings with CC images.	n.d.	DOI 10.7717/peerj-cs.3332/table-4 ( <a href="#">open</a> )
[Peseux et al., 2023]	Stochastic Gradient Descent with Gradient Estimator for Categorical Features	2023	DOI 10.2139/ssrn.4439301 ( <a href="#">open</a> )
[Theodoridis, 2015]	Stochastic Gradient Descent	2015	DOI 10.1016/b978-0-12-801522-3.00005-7 ( <a href="#">open</a> )
[Cambridge University Press, 2014]	Stochastic Gradient Descent	2014	DOI 10.1017/cbo9781107298019.015 ( <a href="#">open</a> )
[SPIE-Intl Soc Optical Eng, a]	10.1117/12.2305101.5783306294001	n.d.	DOI 10.1117/12.2305101.5783306294001 ( <a href="#">open</a> )
[Lan, 2023]	Stochastic Gradient Descent	2023	DOI 10.1007/978-3-030-54621-2_777-1 ( <a href="#">open</a> )
[SPIE-Intl Soc Optical Eng, b]	10.1117/12.2512817.6013939792001	n.d.	DOI 10.1117/12.2512817.6013939792001 ( <a href="#">open</a> )
[SPIE-Intl Soc Optical Eng, e]	10.1117/12.2643564.6314793833112	n.d.	DOI 10.1117/12.2643564.6314793833112 ( <a href="#">open</a> )
[Song]	An Angle-based Stochastic Gradient Descent Method for Machine Learning: Principle and Application	n.d.	DOI 10.25148/etd.fidc009555 ( <a href="#">open</a> )
[Turali and Kozat, 2024]	Optimal Stochastic Gradient Descent Algorithm for Filtering	2024	DOI 10.2139/ssrn.4879640 ( <a href="#">open</a> )
[Ketkar, 2017]	Stochastic Gradient Descent	2017	DOI 10.1007/978-1-4842-2766-4_8 ( <a href="#">open</a> )
[Alonso, 2025]	The Mathematics of Stochastic Gradient Descent and Non-Convex Optimization	2025	DOI 10.2139/ssrn.5032378 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[Pillaud-Vivien]	Learning with reproducing kernel Hilbert spaces : stochastic gradient descent and laplacian estimation	n.d.	DOI 10.70675/f938eaabz37fcz4483zba85z57f0 ( <a href="#">open</a> )
[Amor et al., 2024]	Real-Time Traffic Prediction Through Stochastic Gradient Descent	2024	DOI 10.5220/0012687400003702 ( <a href="#">open</a> )
[Yang and Ma, 2022]	Adaptive stochastic gradient descent for large-scale learning problems	2022	DOI 10.21203/rs.3.rs-1066512/v1 ( <a href="#">open</a> )
[Arefin and Asadujjaman, 2016]	Minimizing Average of Loss Functions Using Gradient Descent and Stochastic Gradient Descent	2016	DOI 10.3329/dujs.v64i2.54490 ( <a href="#">open</a> )
[Chen]	Distributed Stochastic Gradient Descent with Staleness: A Stochastic Delay Differential Equation Based Framework_supp1-3546574.pdf	n.d.	DOI 10.1109/tsp.2025.3546574/mm1 ( <a href="#">open</a> )
[Abubaker et al., 2022a]	Scaling Stratified Stochastic Gradient Descent for Distributed Matrix Completion	2022	DOI 10.36227/tehrxiv.19350536.v1 ( <a href="#">open</a> )
[Fu and Foondun, 2025]	A Theoretical and Experimental Study of Gradient Descent and Its Stochastic Variants	2025	DOI 10.36227/tehrxiv.176162233.33076025/ ( <a href="#">open</a> )
[Abubaker et al., 2022b]	Scaling Stratified Stochastic Gradient Descent for Distributed Matrix Completion	2022	DOI 10.36227/tehrxiv.19350536 ( <a href="#">open</a> )
[SPIE-Intl Soc Optical Eng, g]	10.1117/12.3028105.e628af96-d7c5-ee11-a99e-00505691c5e1	n.d.	DOI 10.1117/12.3028105.e628af96-d7c5-ee11-a99e-00505691c5e1 ( <a href="#">open</a> )
[Tarkhan and Simon, 2020]	bigSurvSGD: Big Survival Analysis Using Stochastic Gradient Descent	2020	DOI 10.32614/cran.package.bigSurvsgd ( <a href="#">open</a> )
[Lam and Wang, 2023]	Resampling Stochastic Gradient Descent Cheaply	2023	DOI 10.1109/wsc60868.2023.10408023 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[Sun et al., 2023]	Noisy Stochastic Gradient Descent Algorithm with Stochastic Event-Triggered Mechanism for Communication-Efficient Distributed Learning	2023	DOI 10.2139/ssrn.4588764 ( <a href="#">open</a> )
[Theodoridis, 2020]	Online Learning: the Stochastic Gradient Descent Family of Algorithms	2020	DOI 10.1016/b978-0-12-818803-3.00014-3 ( <a href="#">open</a> )
[Theodoridis, 2026]	Online learning: the stochastic gradient descent family of algorithms	2026	DOI 10.1016/b978-0-44-329238-5.00011-1 ( <a href="#">open</a> )
[Ali, 2023]	Comparing the Effectiveness of Support Vector Classifier and Stochastic Gradient Descent in Hate-Speech Detection	2023	DOI 10.47611/harp.315 ( <a href="#">open</a> )
[Guo, 2022]	Variable selection of regularized stochastic gradient descent in logistic regression	2022	DOI 10.54647/mathematics11319 ( <a href="#">open</a> )
[Cacciola et al., 2023]	On the Convergence of Stochastic Gradient Descent in Low-Precision Number Formats	2023	DOI 10.5220/0011795500003411 ( <a href="#">open</a> )
[Sirignano and Spiliopoulos, 2017]	Stochastic Gradient Descent in Continuous Time	2017	DOI 10.2139/ssrn.2954149 ( <a href="#">open</a> )
[Abrolbekov, 2026]	High Probability Convergence Bounds for Non-convex Stochastic Gradient Descent with Sub-Weibull Noise	2026	DOI 10.2139/ssrn.6738058 ( <a href="#">open</a> )
[SPIE-Intl Soc Optical Eng, d]	10.1117/12.2599744.6269411057001	n.d.	DOI 10.1117/12.2599744.6269411057001 ( <a href="#">open</a> )
[Nguyen]	Heavy-tailed nature of stochastic gradient descent in deep learning : theoretical and empirical analysis	n.d.	DOI 10.70675/674d4873zb009z425cz9c8ezb79 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[Shrimali et al., 2023]	A comparative study of gradient descent and stochastic gradient descent method for optimization	2023	DOI 10.1063/5.0148736 (open)
[{Chapman and Hall/CRC}, 2011]	Large-Scale Machine Learning with Stochastic Gradient Descent	2011	DOI 10.1201/b11429-6 (open)
[Cambridge University Press, 2021e]	Léon Bottou Stochastic Gradient Descent	2021	DOI 10.1017/9781108860604.006 (open)
[Yamada, 2018]	Hyperparameter-free optimizer of stochastic gradient descent that incorporates unit correction and moment estimation	2018	DOI 10.1101/348557 (open)
[Li, 2024c]	Stochastic Gradient Descent in Nonconvex Optimization: Continuous-Time Dynamics and the Role of Learning Rates	2024	DOI 10.2139/ssrn.4980990 (open)
[Cai et al., 2023]	Communication-Efficient Distributed Stochastic Gradient Descent with Pooling Operator	2023	DOI 10.2139/ssrn.4327869 (open)
[Ayadi and Turinici, 2021]	Stochastic Runge-Kutta methods and adaptive SGD-G2 stochastic gradient descent	2021	DOI 10.1109/icpr48806.2021.9412831 (open)
[Chen et al., 2026]	Fractional-order dynamics driven stochastic gradient descent method with momentum	2026	DOI 10.2139/ssrn.6902318 (open)
[Srinivasan, 2026]	An Intelligent Predictive CPU Scheduling Algorithm Using Online Multiple Linear Regression and Stochastic Gradient Descent	2026	DOI 10.2139/ssrn.6956552 (open)

Cite	Title	Year	DOI / URL
[Kour et al., 2024]	Stochastic Gradient Descent Optimized Efficient Transfer Learning Architecture for Brain Tumor Segmentation	2024	DOI 10.2139/ssrn.5012382 ( <a href="#">open</a> )
[Ramos and Ott, 2015]	Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent	2015	DOI 10.15607/rss.2015.xi.002 ( <a href="#">open</a> )
[Sirignano and Spiliopoulos, 2020]	Stochastic Gradient Descent in Continuous Time: A Central Limit Theorem	2020	DOI 10.1287/stsy.2019.0050 ( <a href="#">open</a> )
[Hafshejani et al., 2023]	Fast Armijo line search for stochastic gradient descent	2023	DOI 10.21203/rs.3.rs-2285238/v1 ( <a href="#">open</a> )
[Bottou, 2010]	Large-Scale Machine Learning with Stochastic Gradient Descent	2010	DOI 10.1007/978-3-7908-2604-3_16 ( <a href="#">open</a> )
[Chen et al., 2025]	Revisit Stochastic Gradient Descent for Strongly Convex Objectives: Tight Uniform-in-Time Bounds	2025	DOI 10.2139/ssrn.5546711 ( <a href="#">open</a> )
[Cheng et al., 2025]	Convergence Analysis of the Last Iterate in Distributed Stochastic Gradient Descent with Momentum	2025	DOI 10.2139/ssrn.5351413 ( <a href="#">open</a> )
[Hu]	Adaptive Batch Size Time Evolving Stochastic Gradient Descent for Federated Learning_supp1-3610169.pdf	n.d.	DOI 10.1109/tpami.2025.3610169/mm1 ( <a href="#">open</a> )
[ichi Amari, 1993]	Backpropagation and stochastic gradient descent method	1993	DOI 10.1016/0925-2312(93)90006-o ( <a href="#">open</a> )
[Schraudolph, 1999]	Local gain adaptation in stochastic gradient descent	1999	DOI 10.1049/cp:19991170 ( <a href="#">open</a> )
[Anonymous, 2025]	Noise balance and stationary distribution of stochastic gradient descent	2025	DOI 10.1103/zjq6-nzwd ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[Liu et al., 2024]	Optimizing Synchronous Stochastic Gradient Descent with Local Efficient Sign and Model Averaging Correction	2024	DOI 10.2139/ssrn.4965637 ( <a href="#">open</a> )
[Ito and Onoue, 2026]	Constrained Graph Drawing by Stochastic Gradient Descent	2026	DOI 10.1109/pacificvis68791.2026.00006 ( <a href="#">open</a> )
[Chang et al., 2020]	Efficient phase-locking of 60 fiber lasers by stochastic parallel gradient descent algorithm	2020	DOI 10.3788/col202018.101403 ( <a href="#">open</a> )
[Bao and Maier, 2020]	Stochastic gradient descent algorithm for stochastic optimization in solving analytic continuation problems	2020	DOI 10.3934/fods.2020001 ( <a href="#">open</a> )
[Li, 2024a]	WITHDRAWN: Bridging Stochastic Gradient Descent and Markov Chains: Constant Step-Size Convergence and Richardson-Romberg Extrapolation	2024	DOI 10.21203/rs.3.rs-5202271/v1 ( <a href="#">open</a> )
[Livni, 2024]	The Sample Complexity of Gradient Descent in Stochastic Convex Optimization	2024	DOI 10.52202/079017-2048 ( <a href="#">open</a> )
[Li, 2024b]	WITHDRAWN: Bridging Stochastic Gradient Descent and Markov Chains: Constant Step-Size Convergence and Richardson-Romberg Extrapolation	2024	DOI 10.21203/rs.3.rs-5202271/v2 ( <a href="#">open</a> )
[Koutsibella and Koutroumbas, 2020]	Stochastic gradient descent possibilistic clustering	2020	DOI 10.1145/3411408.3411436 ( <a href="#">open</a> )
[Chen and Wang, 2012]	Dictionary learning with weighted stochastic gradient descent	2012	DOI 10.1109/ic-cps.2012.6384229 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[T and V, 2024]	Hybrid Optimization Model Integrating Gradient Descent and Stochastic Descent for Enhanced Osteoporosis and Osteopenia Recognition	2024	DOI 10.53759/7669/jmc202404032 (open)
[Lee, 2017]	Differentially Private Variance Reduced Stochastic Gradient Descent	2017	DOI 10.1109/ictcs.2017.60 (open)
[SPIE-Intl Soc Optical Eng, c]	10.1117/12.2525953.6062680904001	n.d.	DOI 10.1117/12.2525953.6062680904001 (open)
[Ravasi et al., 2022]	Multi-Dimensional Deconvolution with Stochastic Gradient Descent	2022	DOI 10.3997/2214-4609.202210234 (open)
[Hu et al., 2022]	Efficiency Ordering of Stochastic Gradient Descent	2022	DOI 10.52202/068431-1155 (open)
[AZIMJONOV and Kim, 2023]	Stochastic Gradient Descent Classifier-Based Lightweight Intrusion Detection Systems Using the Most Efficient Feature Subsets of Datasets	2023	DOI 10.2139/ssrn.4378339 (open)
[Sharma, 2021]	Guided parallelized stochastic gradient descent for delay compensation	2021	DOI 10.1016/j.asoc.2021.107084 (open)
[Yaghoubi and Fainekos, 2017]	Hybrid approximate gradient and stochastic descent for falsification of nonlinear systems	2017	DOI 10.23919/acc.2017.7963007 (open)
[Data and Diggavi, 2020]	On Byzantine-Resilient High-Dimensional Stochastic Gradient Descent	2020	DOI 10.1109/isit44484.2020.9174363 (open)
[Cheng et al., 2019]	Static & Dynamic Appointment Scheduling with Stochastic Gradient Descent	2019	DOI 10.23919/acc.2019.8814666 (open)

Cite	Title	Year	DOI / URL
[SPIE-Intl Soc Optical Eng, f]	10.1117/12.3026153.0df877b5-f9b8-ee11-a99d-c49c781f4d15	n.d.	DOI 10.1117/12.3026153.0df877b5-f9b8-ee11-a99d-c49c781f4d15 ( <a href="#">open</a> )
[Song et al., 2021]	AG-SGD: Angle-Based Stochastic Gradient Descent	2021	DOI 10.1109/access.2021.3055993 ( <a href="#">open</a> )
[Jain and Krishnamurthy, 2025]	Controlling stochastic gradient descent using stochastic approximation for robust distributed optimization	2025	DOI 10.3934/naco.2024041 ( <a href="#">open</a> )
[Williams et al., 2022]	Stochastic Gradient Descent for Optimization of Nuclear Systems	2022	DOI 10.21203/rs.3.rs-2073277/v1 ( <a href="#">open</a> )
[Chapman and Hall/CRC], 2014]	On the Convergence Rate of Stochastic Gradient Descent for Strongly Convex Functions	2014	DOI 10.1201/b17558-10 ( <a href="#">open</a> )
[Zamora and Sossa, 2016]	Dendrite morphological neurons trained by stochastic gradient descent	2016	DOI 10.1109/ssci.2016.7849933 ( <a href="#">open</a> )
[Liu et al., 2021]	A Diffusion Approximation Theory of Momentum Stochastic Gradient Descent in Nonconvex Optimization	2021	DOI 10.1287/stsy.2021.0083 ( <a href="#">open</a> )
[Hua et al., 2023]	Machine-Learning Topology Optimization with Stochastic Gradient Descent Optimizer for Heat Conduction Problems	2023	DOI 10.2139/ssrn.4594476 ( <a href="#">open</a> )
[Sun]	On the Decentralized Stochastic Gradient Descent with Markov Chain Sampling_supp1-3297053.pdf	n.d.	DOI 10.1109/tsp.2023.3297053/mm1 ( <a href="#">open</a> )
[Bhardwaj and Cong, 2016]	Practical Efficiency of Asynchronous Stochastic Gradient Descent	2016	DOI 10.1109/mlhpc.2016.010 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[Iutzeler et al., 2024]	Derivatives of Stochastic Gradient Descent in parametric optimization	2024	DOI 10.52202/079017-3775 (open)
[Jiao and Keller-Ressel, 2024]	Emergence of heavy tails in homogenized stochastic gradient descent	2024	DOI 10.52202/079017-0450 (open)
[Cui and Picheny, 2019]	Acoustic Model Optimization Based on Evolutionary Stochastic Gradient Descent with Anchors for Automatic Speech Recognition	2019	DOI 10.21437/interspeech.2019-2620 (open)
[Ferrarotti and Bemporad, 2019]	Synthesis of Optimal Feedback Controllers from Data via Stochastic Gradient Descent	2019	DOI 10.23919/ecc.2019.8796130 (open)
[Wang and Olson, 2014]	Robust pose graph optimization using stochastic gradient descent	2014	DOI 10.1109/icra.2014.6907482 (open)
[Wijnhoven and de With, 2010]	Fast Training of Object Detection Using Stochastic Gradient Descent	2010	DOI 10.1109/icpr.2010.112 (open)
[Sharma, 2018]	Guided Stochastic Gradient Descent Algorithm for inconsistent datasets	2018	DOI 10.1016/j.asoc.2018.09.038 (open)
[ITMO University, 2020]	Testing of the stochastic parallel gradient descent algorithm to the alignment of a two-mirror telescope	2020	DOI 10.17586/1023-5086-2020-87-05-31-41 (open)
[Koren et al., 2022]	Benign Underfitting of Stochastic Gradient Descent	2022	DOI 10.52202/068431-1425 (open)
[Afshar et al., 2009]	Gradient descent optimisation for ILC-based stochastic distribution control	2009	DOI 10.1109/icca.2009.5410612 (open)
[Cambridge University Press, 2025k]	Neural Networks, Backpropagation, and Stochastic Gradient Descent	2025	DOI 10.1017/9781009509435.009 (open)

Cite	Title	Year	DOI / URL
[Li et al., 2021]	Fast Distributed Stochastic Nesterov Gradient Descent Algorithm for Image Classification	2021	DOI 10.1109/cac53003.2021.9727635 (open)
[Mukhopadhyay, 2020]	Stochastic gradient descent for linear systems with sequential matrix entry accumulation	2020	DOI 10.1016/j.sigpro.2020.107494 (open)
[Christensen and Kallsen, 2024]	Is Learning in Biological Neural Networks Based on Stochastic Gradient Descent? An Analysis Using Stochastic Processes	2024	DOI 10.1162/neco_a_01668 (open)
[Archibald et al., 2020]	A Stochastic Gradient Descent Approach for Stochastic Optimal Control	2020	DOI 10.4208/ea-jam.190420.200420 (open)

### 9.3 reproducible research

*Papers retrieved: 100 · per-source contributions: arxiv=100, crossref=100 · backend errors: none*

Cite	Title	Year	DOI / URL
[Gandrud, 2018c]	Introducing Reproducible Research	2018	DOI 10.1201/9781315382548-1 (open)
[Gandrud, 2020c]	Introducing Reproducible Research	2020	DOI 10.1201/9780429031854-2 (open)
[Gandrud, 2018b]	Getting Started with Reproducible Research	2018	DOI 10.1201/9781315382548-2 (open)
[Gandrud, 2020b]	Getting Started with Reproducible Research	2020	DOI 10.1201/9780429031854-3 (open)
[Hoefling and Rossini, 2018]	Reproducible Research for Large-Scale Data Analysis	2018	DOI 10.1201/9781315373461-8 (open)
[Xie, 2018]	knitr: A Comprehensive Tool for Reproducible Research in R	2018	DOI 10.1201/9781315373461-1 (open)
[Baker, 2022]	Reproducible data analysis	2022	DOI 10.1093/hesc/9780192896599.003.0019 (open)

Cite	Title	Year	DOI / URL
[PubMed]	NinBioinformatics Reproducible Research Reports	n.d.	DOI 10.21428/3c290898 ( <a href="#">open</a> )
[University of California Press, 2019c]	Reproducible Workflow	2019	DOI 10.2307/j.ctvvpb3xkg.16 ( <a href="#">open</a> )
[Preeyanon et al., 2018]	Reproducible Bioinformatics Research for Biologists	2018	DOI 10.1201/9781315373461- 7 ( <a href="#">open</a> )
[Davison et al., 2018]	Sumatra: A Toolkit for Reproducible Research	2018	DOI 10.1201/9781315373461- 3 ( <a href="#">open</a> )
[University of California Press, 2019a]	ELEVEN. Reproducible Workflow	2019	DOI 10.1525/9780520969230- 014 ( <a href="#">open</a> )
[Hrynaszkiewicz et al., 2018]	Open Science and the Role of Publishers in Reproducible Research	2018	DOI 10.1201/9781315373461- 15 ( <a href="#">open</a> )
[PeerJ, c]	Supplemental Information 1: Reproducible Research Instructions.	n.d.	DOI 10.7717/peerj- cs.904/supp-1 ( <a href="#">open</a> )
[Stodden, 2014]	Implementing Reproducible Research	2014	DOI 10.1201/b16868 ( <a href="#">open</a> )
[F1000 Research Ltd]	Reproducible Research Data and Software	n.d.	DOI 10.12688/f1000research.channels.908 ( <a href="#">open</a> )
[Kedron et al., 2023]	Reproducible Research Practices and Barriers to Reproducible Research in Geography: Insights from a Survey	2023	DOI 10.31219/osf.io/nyrq9 ( <a href="#">open</a> )
[Murray-Rust and Murray-Rust, 2018]	Reproducible Physical Science and the Declaratron	2018	DOI 10.1201/9781315373461- 5 ( <a href="#">open</a> )
[Burgess, 2018]	Reproducible research article collection	2018	DOI 10.14293/s2199- 1006.1.sor- uncat.clsuuhc.v1 ( <a href="#">open</a> )
[Solt and Hu, 2016]	pewdata: Reproducible Retrieval of Pew Research Center Datasets	2016	DOI 10.32614/cran.package.pewdata ( <a href="#">open</a> )
[Basu, 2017]	Reproducible research with jupyter notebooks	2017	DOI 10.22541/au.151460905.57485984 ( <a href="#">open</a> )
[Edmunds, 2016a]	Reproducible Research Resources for Research(ing) Parasites	2016	DOI 10.59350/63nv3-fa097 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[Edmunds, 2016b]	Reproducible Research Resources for Research(ing) Parasites	2016	DOI 10.59350/vhpgg-ec668 (open)
[Hector, 2021]	Reproducible Research	2021	DOI 10.1093/oso/9780198798170.003.0004 (open)
[Edmunds, 2015a]	Fermenting a Reproducible Research Revolution	2015	DOI 10.59350/ejn9k-s7c54 (open)
[Kunisato, 2019]	Introduction to reproducible psychological research	2019	DOI 10.31234/osf.io/x8js5 (open)
[Suber, 2008a]	OA for reproducible research	2008	DOI 10.63485/4jjac-1dd97 (open)
[Hinsen, 2013a]	Platforms for reproducible research	2013	DOI 10.59350/s56jr-3cn95 (open)
[Edmunds, 2015b]	Fermenting a Reproducible Research Revolution	2015	DOI 10.59350/0hs8b-m2239 (open)
[Turek and Deniz, 2019]	Case Studies in Reproducible Research	2019	DOI 10.1525/9780520967779-006 (open)
[Gandrud, 2020d]	Reproducible Research with R and RStudio	2020	DOI 10.1201/9780429031854 (open)
[Basu, 2018]	Reproducible research using minimalist plain text tools	2018	DOI 10.32388/649864 (open)
[Charlton, 2016]	How do we ensure that research is reproducible?	2016	DOI 10.15200/winn.146397.78741 (open)
[Gandrud, 2018d]	Reproducible Research with R and RStudio	2018	DOI 10.1201/9781315382548 (open)
[Bahlai, 2016]	The open science and reproducible research course	2016	DOI 10.7490/f1000research.1112876.1 (open)
[Gandrud, 2013]	Reproducible Research with R and R Studio	2013	DOI 10.1201/b15100 (open)
[Edmunds, 2014a]	CARMEN, reproducible research and push-button papers	2014	DOI 10.59350/xx4rg-zcd90 (open)
[Kitzes, 2019]	The Basic Reproducible Workflow Template	2019	DOI 10.1525/9780520967779-005 (open)
[MISSING-VALUE, a]	Platforms	n.d.	DOI 10.1201/b16868-20 (open)

Cite	Title	Year	DOI / URL
[White, 2018]	Software skills for reproducible data-intensive research	2018	DOI 10.7490/f1000research.1115901.1 ( <a href="#">open</a> )
[Weisbrod, 2026]	example-project: A Reproducible Empirical Research Template	2026	DOI 10.31235/osf.io/yx7af_v1 ( <a href="#">open</a> )
[SAGE Publications Ltd, 2020]	Transparent and Reproducible Data Analysis	2020	DOI 10.4135/9781526421036926480 ( <a href="#">open</a> )
[Suber, 2008b]	Open licensing to enable reproducible research	2008	DOI 10.63485/8k73p-vrh09 ( <a href="#">open</a> )
[Chapman and Hall/CRC], 2016a]	Reproducible Research with R and R Studio, Second Edition	2016	DOI 10.1201/b18546 ( <a href="#">open</a> )
[MISSING-VALUE, c]	Tools	n.d.	DOI 10.1201/b16868-6 ( <a href="#">open</a> )
[Edmunds, 2014b]	CARMEN, reproducible research and push-button papers	2014	DOI 10.59350/p6xyp-h2p77 ( <a href="#">open</a> )
[PeerJ, b]	Figure 2: The final reproducible research criteria used for the evaluation.	n.d.	DOI 10.7717/peerj.5072/fig-2 ( <a href="#">open</a> )
[Sage Publications, Ltd., 2025]	Preparing for Transparent and Reproducible Quantitative Social Science Research	2025	DOI 10.4135/9781036230647 ( <a href="#">open</a> )
[Hinsen, 2016]	From reproducible to verifiable computer-aided research	2016	DOI 10.59350/fj0g3-5yv59 ( <a href="#">open</a> )
[Suber, 2009]	More on OA to facilitate reproducible research	2009	DOI 10.63485/tys9t-3y862 ( <a href="#">open</a> )
[Dutch Research Council (NWO)]	10.61686/udeac59432	n.d.	DOI 10.61686/udeac59432 ( <a href="#">open</a> )
[Geert and Koßmann]	Reproducible research reports with Quarto	n.d.	DOI 10.21428/1192f2f8.fb1a1b7f ( <a href="#">open</a> )
[Chapman and Hall/CRC], 2016b]	Reproducible Research	2016	DOI 10.1201/b15166-10 ( <a href="#">open</a> )
[Chapman and Hall/CRC], 2018]	Implementing Reproducible Research	2018	DOI 10.1201/9781315373461 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[Heston, 2023]	Statistics, Ethics, and the Promotion of Reproducible Research	2023	DOI 10.22541/au.169627960.06669688/v1 ( <a href="#">open</a> )
[Moresi, 2018a]	Alaska Moho Model (Reproducible research with containers)	2018	DOI 10.59350/pn8gh-98592 ( <a href="#">open</a> )
[Hinsen, 2013b]	Python as a platform for reproducible research	2013	DOI 10.59350/85pnj-bak53 ( <a href="#">open</a> )
[Wiebels and Moreau, 2021]	Leveraging Containers for Reproducible Psychological Research	2021	DOI 10.31234/osf.io/h7tkg ( <a href="#">open</a> )
[Marwick, 2019]	Case Study 12: Using R and Related Tools for Reproducible Research in Archaeology	2019	DOI 10.1525/9780520967779-021 ( <a href="#">open</a> )
[Roškar, 2022]	Renku: a platform for collaborative, reusable and reproducible research	2022	DOI 10.5194/egusphere-egu22-10697 ( <a href="#">open</a> )
[Gandrud, 2020a]	Conclusion	2020	DOI 10.1201/9780429031854-17 ( <a href="#">open</a> )
[Maerz, 2024]	Quarto in RStudio: Writing Reproducible & Dynamic Research Papers	2024	DOI 10.61700/pdeaaefq62n681556 ( <a href="#">open</a> )
[eLife Sciences Publications, Ltd, 2021]	Decision letter: Is preclinical research in cancer biology reproducible enough?	2021	DOI 10.7554/elife.67527.sa1 ( <a href="#">open</a> )
[Hediyeh-Zadeh and Davis, 2016]	Computational workflows for research students: towards a reproducible research	2016	DOI 10.7490/f1000research.1113332.1 ( <a href="#">open</a> )
[Gandrud, 2018a]	Conclusion	2018	DOI 10.1201/9781315382548-14 ( <a href="#">open</a> )
[MISSING-VALUE, b]	Practices and Guidelines	n.d.	DOI 10.1201/b16868-12 ( <a href="#">open</a> )
[Limare]	Reproducible research, software quality, online interfaces and publishing for image processing	n.d.	DOI 10.70675/d17fee18z3990z46f0z8820zb7fa ( <a href="#">open</a> )
[Moresi, 2018b]	Alaska Moho Model (Reproducible research with containers)	2018	DOI 10.59350/15qd1-96r10 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[Boettiger, 2019]	Case Study 3: A Reproducible R Notebook Using Docker	2019	DOI 10.1525/9780520967779-012 ( <a href="#">open</a> )
[Solt and Gracey, 2016]	icpsrdata: Reproducible Data Retrieval from the ICPSR Archive	2016	DOI 10.32614/cran.package.icpsrdata ( <a href="#">open</a> )
[Peyrache, 2026]	eLife Assessment: Spyglass: a framework for reproducible and shareable neuroscience research	2026	DOI 10.7554/elife.108089.2.sa3 ( <a href="#">open</a> )
[eLife Sciences Publications, Ltd, 2025a]	Reviewer #1 (Public review): Spyglass: a framework for reproducible and shareable neuroscience research	2025	DOI 10.7554/elife.108089.1.sa1 ( <a href="#">open</a> )
[Gavel, 2025]	Ensuring Your Quantitative Research is Replicable and Reproducible	2025	DOI 10.4135/9781036216696 ( <a href="#">open</a> )
[Hinsen, 2017b]	Sustainable software and reproducible research: dealing with software collapse	2017	DOI 10.59350/7tavk-2hf75 ( <a href="#">open</a> )
[Basu, 2021]	How to use orgmode for reproducible research, rough cut version 1	2021	DOI 10.32388/b5rvo7 ( <a href="#">open</a> )
[Turner, 2010a]	Sweave for Reproducible Research and Beautiful Statistical Reports	2010	DOI 10.59350/247k5-kfv86 ( <a href="#">open</a> )
[Turner, 2009a]	Seminar: Reproducible Research with R, LaTeX, & Sweave	2009	DOI 10.59350/2958f-vdz74 ( <a href="#">open</a> )
[Peyrache, 2025]	eLife Assessment: Spyglass: a framework for reproducible and shareable neuroscience research	2025	DOI 10.7554/elife.108089.1.sa2 ( <a href="#">open</a> )
[LeBeau et al., 2020]	Reproducible Analyses in Educational Research	2020	DOI 10.17077/pp.005637 ( <a href="#">open</a> )
[Yildiz and Kowalski, 2023]	Data-integrated executable publications for reproducible geohazards research	2023	DOI 10.5194/egusphere-egu23-7417 ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[{Springer Science and Business Media LLC}, 2017]	Fostering reproducible fMRI research	2017	DOI 10.1038/ncomms14748 ( <a href="#">open</a> )
[eLife Sciences Publications, Ltd, 2026a]	Reviewer #2 (Public review): Spyglass: a framework for reproducible and shareable neuroscience research	2026	DOI 10.7554/elife.108089.2.sa1 ( <a href="#">open</a> )
[eLife Sciences Publications, Ltd, 2025b]	Reviewer #2 (Public review): Spyglass: a framework for reproducible and shareable neuroscience research	2025	DOI 10.7554/elife.108089.1.sa0 ( <a href="#">open</a> )
[Chow, 2019]	Reproducible Research	2019	DOI 10.1201/9780429275067-9 ( <a href="#">open</a> )
[Alston and Rick, 2020]	A Beginner’s Guide to Conducting Reproducible Research	2020	DOI 10.32942/osf.io/h5r6n ( <a href="#">open</a> )
[Hinsen, 2017a]	Reproducible research in the Python ecosystem: a reality check	2017	DOI 10.59350/wgebd-68y68 ( <a href="#">open</a> )
[Turner, 2010b]	Sweave for Reproducible Research and Beautiful Statistical Reports	2010	DOI 10.59350/pexqx-j5g86 ( <a href="#">open</a> )
[Hastings, 2023]	AI for Reproducible Research	2023	DOI 10.1201/9781003226642-4 ( <a href="#">open</a> )
[eLife Sciences Publications, Ltd, 2026b]	Reviewer #1 (Public review): Spyglass: a framework for reproducible and shareable neuroscience research	2026	DOI 10.7554/elife.108089.2.sa2 ( <a href="#">open</a> )
[Strand and Brown, 2019]	Publishing open, reproducible research with undergraduates	2019	DOI 10.31234/osf.io/f7kuy ( <a href="#">open</a> )
[Hinsen, 2012]	Unifying version control and dependency management for reproducible research	2012	DOI 10.59350/986d5-0he30 ( <a href="#">open</a> )
[Mittal, 2025]	Explainable AI-Augmented DevSecOps for Secure and Reproducible Cloud-Native Research Software	2025	DOI 10.36227/techriv.175617187.78775647/ ( <a href="#">open</a> )

Cite	Title	Year	DOI / URL
[Ohta and Ogasawara, 2015]	Container-based sequence data analysis workflow for reproducible research	2015	DOI 10.7490/f1000research.1110170.1 ( <a href="#">open</a> )
[Turner, 2009b]	Seminar: Reproducible Research with R, LaTeX, & Sweave	2009	DOI 10.59350/d3cd0-5jy93 ( <a href="#">open</a> )
[Butland, 2019]	Community Call - Reproducible Research with R	2019	DOI 10.59350/v7xr3-6sn63 ( <a href="#">open</a> )
[Baggerly, 2024]	The Importance of Reproducible Research in High-Throughput Biology	2024	DOI 10.52519/00193 ( <a href="#">open</a> )
[University of California Press, 2019b]	Introduction	2019	DOI 10.2307/j.ctvpb3xkg.6 ( <a href="#">open</a> )
[University of California Press, 2019d]	Tables	2019	DOI 10.1525/9780520969230-002 ( <a href="#">open</a> )
[Poldrack, 2019]	Case Study 29: Developing a Reproducible Workflow for Large-Scale Phenotyping	2019	DOI 10.1525/9780520967779-038 ( <a href="#">open</a> )
[University of California Press, 2019e]	What Is Ethical Research?	2019	DOI 10.2307/j.ctvpb3xkg.7 ( <a href="#">open</a> )

*Per-paper synthesis omitted — no LLM Contribution paragraphs are present in the deep-search outputs (set `deep_search.llm_per_paper: true` and ensure Ollama is reachable to populate this section).*

*Composition summary: 3 keywords · 300 unique papers · 300 per-paper notes integrated · 300 BibTeX entries · 0 key(s) missing from bib.*

## 10 References

This project can produce **two** BibTeX files; the template combined-PDF path uses Pandoc `--natbib` plus BibTeX and merges every `manuscript/*.bib` for citation resolution:

- `manuscript/references.bib` — single-query pipeline output (`scripts/run_search_pipeline.py`).
- `manuscript/references_deep.bib` — deduplicated multi-keyword deep-search output (`scripts/run_deep_search.py`). Every citation in sec. 9 resolves against this file. The supplemental section is auto-composed by `scripts/s_compose_literature_review.py`; do not edit by hand.

To regenerate the standard bibliography:

```
uv run python projects/templates/template_search_project/scripts/run_search_pipeline.py
```

To regenerate the deep-search bibliography (10 papers per keyword, fully enriched, LLM-summarised):

```
uv run python projects/templates/template_search_project/scripts/run_deep_search.py
```

```
uv run python projects/templates/template_search_project/scripts/s_compose_literature_review.py
```

To validate that either `.bib` is syntactically clean and contains the required fields per entry type:

```
uv run python -m infrastructure.reference.citation.cli validate \  
  projects/templates/template_search_project/manuscript/references.bib --strict  
uv run python -m infrastructure.reference.citation.cli validate \  
  projects/templates/template_search_project/manuscript/references_deep.bib --strict
```

## References

- M. Hirschberger \*. Connectedness of efficient points in convex and convex transformable vector optimization. *Optimization*, 54(3):283–304, 2005. doi: 10.1080/02331930500096270.
- Makhamadamin Abrolbekov. High probability convergence bounds for non-convex stochastic gradient descent with sub-weibull noise. 2026. doi: 10.2139/ssrn.6738058.
- Nabil Abubaker, M. Ozan Karsavuran, and Cevdet Aykanat. Scaling stratified stochastic gradient descent for distributed matrix completion. 2022a. doi: 10.36227/tehrxiv.19350536.v1.
- Nabil Abubaker, M. Ozan Karsavuran, and Cevdet Aykanat. Scaling stratified stochastic gradient descent for distributed matrix completion. 2022b. doi: 10.36227/tehrxiv.19350536.
- Puya Afshar, Martin Brown, and Hong Wang. Gradient descent optimisation for ilc-based stochastic distribution control. In *2009 IEEE International Conference on Control and Automation*. IEEE, 2009. doi: 10.1109/icca.2009.5410612.
- Dania Ali. Comparing the effectiveness of support vector classifier and stochastic gradient descent in hate-speech detection. 2023. doi: 10.47611/harp.315.
- Miquel Nogueira I Alonso. The mathematics of stochastic gradient descent and non-convex optimization. 2025. doi: 10.2139/ssrn.5032378.
- Jesse Alston and Jessica Rick. A beginner’s guide to conducting reproducible research. 2020. doi: 10.32942/osf.io/h5r6n.
- Yasmine Amor, Lilia Rejeb, Nabil Sahli, Wassim Trojet, Lamjed Ben Said, and Ghaleb Hoblos. Real-time traffic prediction through stochastic gradient descent. In *Proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems*, pages 361–369. SCITEPRESS - Science and Technology Publications, 2024. doi: 10.5220/0012687400003702.
- Anonymous. Noise balance and stationary distribution of stochastic gradient descent. *Physical Review E*, 2025. doi: 10.1103/zjq6-nzwd.
- Richard Archibald, Feng Bao, and Jiongmin Yong. A stochastic gradient descent approach for stochastic optimal control. *East Asian Journal on Applied Mathematics*, 10(4):635–658, 2020. doi: 10.4208/eajam.190420.200420.
- Md Rajib Arefin and M Asadujjaman. Minimizing average of loss functions using gradient descent and stochastic gradient descent. *Dhaka University Journal of Science*, 64(2):141–145, 2016. doi: 10.3329/dujs.v64i2.54490.
- Imen Ayadi and Gabriel Turinici. Stochastic runge-kutta methods and adaptive sgd-g2 stochastic gradient descent. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8220–8227. IEEE, 2021. doi: 10.1109/icpr48806.2021.9412831.
- Jahongir AZIMJONOV and Taehong Kim. Stochastic gradient descent classifier-based lightweight intrusion detection systems using the most efficient feature subsets of datasets. 2023. doi: 10.2139/ssrn.4378339.
- Keith Baggerly. The importance of reproducible research in high-throughput biology. In *IGCT Workshops*. University of Texas MD Anderson Cancer Center, 2024. doi: 10.52519/00193.
- Christine A. Bahlai. The open science and reproducible research course. 2016. doi: 10.7490/fl1000research.1112876.1.
- Daniel H. Baker. *Reproducible data analysis*. Oxford University Press, 2022. ISBN 9780192896599. doi: 10.1093/hesc/9780192896599.003.0019.
- Feng Bao and Thomas Maier. Stochastic gradient descent algorithm for stochastic optimization in solving analytic continuation problems. *Foundations of Data Science*, 2(1):1–17, 2020. doi: 10.3934/fods.2020001.
- Arindam Basu. Reproducible research with jupyter notebooks. 2017. doi: 10.22541/au.151460905.57485984.

- Arindam Basu. Reproducible research using minimalist plain text tools. 2018. doi: 10.32388/649864.
- Arindam Basu. How to use orgmode for reproducible research, rough cut version 1. 2021. doi: 10.32388/b5rvo7.
- Onkar Bhardwaj and Guojing Cong. Practical efficiency of asynchronous stochastic gradient descent. In *2016 2nd Workshop on Machine Learning in HPC Environments (MLHPC)*, pages 56–62. IEEE, 2016. doi: 10.1109/mlhpc.2016.010.
- Carl Boettiger. *Case Study 3: A Reproducible R Notebook Using Docker*. University of California Press, 2019. ISBN 9780520967779. doi: 10.1525/9780520967779-012.
- J. Frédéric Bonnans. *A Convex Optimization Toolbox*. Springer International Publishing, 2019. ISBN 9783030149765. doi: 10.1007/978-3-030-14977-2\_1.
- Léon Bottou. *Large-Scale Machine Learning with Stochastic Gradient Descent*. Physica-Verlag HD, 2010. ISBN 9783790826036. doi: 10.1007/978-3-7908-2604-3\_16.
- Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, 2004. ISBN 978-0-521-83378-3. doi: 10.1017/CBO9780511804441.
- Steven Burgess. Reproducible research article collection. 2018. doi: 10.14293/s2199-1006.1.sor-uncat.csluuhc.v1.
- Stefanie Butland. Community call - reproducible research with r. 2019. doi: 10.59350/v7xr3-6sn63.
- Matteo Cacciola, Antonio Frangioni, Masoud Asgharian, Alireza Ghaffari, and Vahid Nia. On the convergence of stochastic gradient descent in low-precision number formats. In *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods*, pages 542–549. SCITEPRESS - Science and Technology Publications, 2023. doi: 10.5220/0011795500003411.
- Zhengao Cai, Aiguo Chen, Yi Luo, and Jiahao Li. Communication-efficient distributed stochastic gradient descent with pooling operator. 2023. doi: 10.2139/ssrn.4327869.
- Cambridge University Press. *Background*. Cambridge University Press, a. doi: 10.1017/cbo9781139924672.003.
- Cambridge University Press. *Economics*. Cambridge University Press, b. doi: 10.1017/cbo9781139924672.007.
- Cambridge University Press. *Introduction*. Cambridge University Press, c. doi: 10.1017/cbo9781139924672.002.
- Cambridge University Press. *Preface*. Cambridge University Press, d. doi: 10.1017/cbo9781139924672.001.
- Cambridge University Press. *Approximation and fitting*. Cambridge University Press, 2004a. doi: 10.1017/cbo9780511804441.007.
- Cambridge University Press. *Convex optimization problems*. Cambridge University Press, 2004b. ISBN 9780521833783. doi: 10.1017/cbo9780511804441.005.
- Cambridge University Press. *Convex functions*. Cambridge University Press, 2004c. doi: 10.1017/cbo9780511804441.004.
- Cambridge University Press. *Convex sets*. Cambridge University Press, 2004d. ISBN 9780521833783. doi: 10.1017/cbo9780511804441.003.
- Cambridge University Press. *Duality*. Cambridge University Press, 2004e. ISBN 9780521833783. doi: 10.1017/cbo9780511804441.006.
- Cambridge University Press. *Equality constrained minimization*. Cambridge University Press, 2004f. doi: 10.1017/cbo9780511804441.011.

Cambridge University Press. *Geometric problems*. Cambridge University Press, 2004g. doi: 10.1017/cbo9780511804441.009.

Cambridge University Press. *Introduction*. Cambridge University Press, 2004h. doi: 10.1017/cbo9780511804441.002.

Cambridge University Press. Mathematical background. *Convex Optimization*, pages 631–652, 2004i. doi: 10.1017/cbo9780511804441.013.

Cambridge University Press. *Preface*. Cambridge University Press, 2004j. doi: 10.1017/cbo9780511804441.001.

Cambridge University Press. *References*. Cambridge University Press, 2004k. doi: 10.1017/cbo9780511804441.016.

Cambridge University Press. *Statistical estimation*. Cambridge University Press, 2004l. doi: 10.1017/cbo9780511804441.008.

Cambridge University Press. *Unconstrained minimization*. Cambridge University Press, 2004m. doi: 10.1017/cbo9780511804441.010.

Cambridge University Press. *Stochastic Gradient Descent*. Cambridge University Press, 2014. doi: 10.1017/cbo9781107298019.015.

Cambridge University Press. *Convex Optimization and Efficiency*. Cambridge University Press, 2021a. ISBN 9781108699211. doi: 10.1017/9781108699211.006.

Cambridge University Press. *Convex Optimization*. Cambridge University Press, 2021b. ISBN 9781108980647. doi: 10.1017/9781108980647.012.

Cambridge University Press. Convex analysis and convex optimization. *Compressive Imaging: Structure, Sampling, Learning*, pages 546–552, 2021c. doi: 10.1017/9781108377447.033.

Cambridge University Press. *Ellipsoid Method for Convex Optimization*. Cambridge University Press, 2021d. ISBN 9781108699211. doi: 10.1017/9781108699211.015.

Cambridge University Press. *Stochastic Gradient Descent*. Cambridge University Press, 2021e. ISBN 9781108860604. doi: 10.1017/9781108860604.006.

Cambridge University Press. *Cone-Convex Functions: Elementary Calculus and Examples*. Cambridge University Press, 2025a. ISBN 9781009510561. doi: 10.1017/9781009510561.025.

Cambridge University Press. Convex functions. *Essential Mathematics for Convex Optimization*, pages 151–232, 2025b. doi: 10.1017/9781009510561.012.

Cambridge University Press. *Convex Programming Problems and Convex Theorem of the Alternative*. Cambridge University Press, 2025c. ISBN 9781009510561. doi: 10.1017/9781009510561.021.

Cambridge University Press. Convex programming, lagrange duality, saddle points. *Essential Mathematics for Convex Optimization*, pages 233–341, 2025d. doi: 10.1017/9781009510561.020.

Cambridge University Press. *Convex Programming in Cone-Constrained Form*. Cambridge University Press, 2025e. ISBN 9781009510561. doi: 10.1017/9781009510561.023.

Cambridge University Press. *Convex Optimization*. Cambridge University Press, 2025f. ISBN 9781009493512. doi: 10.1017/9781009493512.006.

Cambridge University Press. Convex optimization theory. *Portfolio Optimization*, pages 491–538, 2025g. doi: 10.1017/9781009428095.021.

Cambridge University Press. *First Acquaintance with Convex Sets*. Cambridge University Press, 2025h. ISBN 9781009510561. doi: 10.1017/9781009510561.003.

- Cambridge University Press. *First Acquaintance with Convex Functions*. Cambridge University Press, 2025i. ISBN 9781009510561. doi: 10.1017/9781009510561.013.
- Cambridge University Press. *Minima and Maxima of Convex Functions*. Cambridge University Press, 2025j. ISBN 9781009510561. doi: 10.1017/9781009510561.015.
- Cambridge University Press. *Neural Networks, Backpropagation, and Stochastic Gradient Descent*. Cambridge University Press, 2025k. ISBN 9781009509435. doi: 10.1017/9781009509435.009.
- Cambridge University Press. *Optimality Conditions in Convex Programming*. Cambridge University Press, 2025l. ISBN 9781009510561. doi: 10.1017/9781009510561.024.
- Cambridge University Press. *Separation Theorem and Geometry of Convex Sets*. Cambridge University Press, 2025m. ISBN 9781009510561. doi: 10.1017/9781009510561.009.
- Hongxiang Chang, Jiachao Xi, Rongtao Su, Pengfei Ma, Yanxing Ma, and Pu Zhou. Efficient phase-locking of 60 fiber lasers by stochastic parallel gradient descent algorithm. *Chinese Optics Letters*, 18(10):101403, 2020. doi: 10.3788/col202018.101403.
- {Chapman and Hall/CRC}. *Large-Scale Machine Learning with Stochastic Gradient Descent* Léon Bottou. Chapman and Hall/CRC, 2011. ISBN 9780429107689. doi: 10.1201/b11429-6.
- {Chapman and Hall/CRC}. *On the Convergence Rate of Stochastic Gradient Descent for Strongly Convex Functions*. Chapman and Hall/CRC, 2014. ISBN 9780429076121. doi: 10.1201/b17558-10.
- {Chapman and Hall/CRC}. *Reproducible Research with R and R Studio, Second Edition*. Chapman and Hall/CRC, 2016a. ISBN 9781498715386. doi: 10.1201/b18546.
- {Chapman and Hall/CRC}. *Reproducible Research*. Chapman and Hall/CRC, 2016b. ISBN 9780429171031. doi: 10.1201/b15166-10.
- {Chapman and Hall/CRC}. *Implementing Reproducible Research*. Chapman and Hall/CRC, 2018. ISBN 9781315373461. doi: 10.1201/9781315373461.
- Bruce Charlton. How do we ensure that research is reproducible? 2016. doi: 10.15200/winn.146397.78741.
- Kang Chen, Yasong Feng, and Tianyu Wang. Revisit stochastic gradient descent for strongly convex objectives: Tight uniform-in-time bounds. 2025. doi: 10.2139/ssrn.5546711.
- Lang Chen and Jianjun Wang. Dictionary learning with weighted stochastic gradient descent. In *2012 International Conference on Computational Problem-Solving (ICCP)*, pages 9–12. IEEE, 2012. doi: 10.1109/iccps.2012.6384229.
- Wei Chen. Distributed stochastic gradient descent with staleness: A stochastic delay differential equation based framework\_supp1-3546574.pdf. doi: 10.1109/tsp.2025.3546574/mm1.
- Yuquan Chen, Hong Wenchao, Mo Xuan, and Bing Wang. Fractional-order dynamics driven stochastic gradient descent method with momentum. 2026. doi: 10.2139/ssrn.6902318.
- Difei Cheng, Ruinan Jin, and Bo Zhang. Convergence analysis of the last iterate in distributed stochastic gradient descent with momentum. 2025. doi: 10.2139/ssrn.5351413.
- Gary Cheng, Kabir Chandrasekher, and Jean Walrand. Static & dynamic appointment scheduling with stochastic gradient descent. In *2019 American Control Conference (ACC)*, pages 2092–2099. IEEE, 2019. doi: 10.23919/acc.2019.8814666.
- Shein-Chung Chow. *Reproducible Research*. Chapman and Hall/CRC, 2019. ISBN 9780429275067. doi: 10.1201/9780429275067-9.
- Sören Christensen and Jan Kallsen. Is learning in biological neural networks based on stochastic gradient descent? an analysis using stochastic processes. *Neural Computation*, 36(7):1424–1432, 2024. doi: 10.1162/neco\_a\_01668.

- CRC Press. *Algorithms for Conic Optimization*. CRC Press, 2011a. ISBN 9780429139833. doi: 10.1201/b10839-12.
- CRC Press. *Cones, Complementarity, and Conic Optimization*. CRC Press, 2011b. ISBN 9780429139833. doi: 10.1201/b10839-5.
- CRC Press. *Convex Semi-Infinite Optimization*. CRC Press, 2011c. ISBN 9780429065026. doi: 10.1201/b11156-17.
- CRC Press. *Tools for Convex Optimization*. CRC Press, 2011d. ISBN 9780429065026. doi: 10.1201/b11156-8.
- CRC Press. *Tools for Convex Optimization*. CRC Press, 2011e. ISBN 9781439868225. doi: 10.1201/b11156-3.
- CRC Press. *Weak Sharp Minima in Convex Optimization*. CRC Press, 2011f. ISBN 9780429065026. doi: 10.1201/b11156-15.
- CRC Press. *What Is Convex Optimization?* CRC Press, 2011g. ISBN 9780429065026. doi: 10.1201/b11156-7.
- CRC Press. *What Is Convex Optimization?* CRC Press, 2011h. ISBN 9781439868225. doi: 10.1201/b11156-2.
- CRC Press. *Convex Optimization Problems*. CRC Press, 2017. ISBN 9781498776455. doi: 10.1201/9781315366920-5.
- Xiaodong Cui and Michael Picheny. Acoustic model optimization based on evolutionary stochastic gradient descent with anchors for automatic speech recognition. In *Interspeech 2019*, pages 1581–1585. ISCA, 2019. doi: 10.21437/interspeech.2019-2620.
- Deepesh Data and Suhas Diggavi. On byzantine-resilient high-dimensional stochastic gradient descent. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2628–2633. IEEE, 2020. doi: 10.1109/isit44484.2020.9174363.
- Andrew P. Davison, Michele Mattioni, Dmitry Samarkanov, and Bartosz Teleńczuk. *Sumatra: A Toolkit for Reproducible Research*. Chapman and Hall/CRC, 2018. ISBN 9781315373461. doi: 10.1201/9781315373461-3.
- DE GRUYTER. *2 Convex sets and convex functions*. DE GRUYTER, 2014a. ISBN 9783110361032. doi: 10.1515/9783110361629.30.
- DE GRUYTER. *6 Convex optimization algorithms*. DE GRUYTER, 2014b. ISBN 9783110361032. doi: 10.1515/9783110361629.117.
- De Gruyter Open Poland. *4 Convex Nonsmooth Optimization*. De Gruyter Open Poland, 2014. doi: 10.2478/9783110426045.4.
- F. Dragomirescu and C. Ivan. The smallest convex extensions of a convex function. *Optimization*, 24(3-4): 193–206, 1992. doi: 10.1080/02331939208843789.
- Dutch Research Council (NWO). 10.61686/udeac59432. doi: 10.61686/udeac59432.
- Joydeep Dutta. Barrier method in nonsmooth convex optimization without convex representation. *Optimization Letters*, 9(6):1177–1185, 2014. doi: 10.1007/s11590-014-0811-1.
- Scott Edmunds. Carmen, reproducible research and push-button papers. 2014a. doi: 10.59350/xx4rg-zcd90.
- Scott Edmunds. Carmen, reproducible research and push-button papers. 2014b. doi: 10.59350/p6xyp-h2p77.
- Scott Edmunds. Fermenting a reproducible research revolution. 2015a. doi: 10.59350/ejn9k-s7c54.
- Scott Edmunds. Fermenting a reproducible research revolution. 2015b. doi: 10.59350/0hs8b-m2239.
- Scott Edmunds. Reproducible research resources for research(ing) parasites. 2016a. doi: 10.59350/63nv3-fa097.
- Scott Edmunds. Reproducible research resources for research(ing) parasites. 2016b. doi: 10.59350/vhpgg-ec668.

- eLife Sciences Publications, Ltd. Decision letter: Is preclinical research in cancer biology reproducible enough? 2021. doi: 10.7554/elife.67527.sa1.
- eLife Sciences Publications, Ltd. Reviewer #1 (public review): Spyglass: a framework for reproducible and shareable neuroscience research. 2025a. doi: 10.7554/elife.108089.1.sa1.
- eLife Sciences Publications, Ltd. Reviewer #2 (public review): Spyglass: a framework for reproducible and shareable neuroscience research. 2025b. doi: 10.7554/elife.108089.1.sa0.
- eLife Sciences Publications, Ltd. Reviewer #2 (public review): Spyglass: a framework for reproducible and shareable neuroscience research. 2026a. doi: 10.7554/elife.108089.2.sa1.
- eLife Sciences Publications, Ltd. Reviewer #1 (public review): Spyglass: a framework for reproducible and shareable neuroscience research. 2026b. doi: 10.7554/elife.108089.2.sa2.
- Elsevier. *Convex Sets. Convex and Generalized Convex Functions*. Elsevier, 2004. ISBN 9780444505507. doi: 10.1016/b978-044450550-7/50002-8.
- Emerald Publishing Limited. *Convex Optimization Basics*. Emerald Publishing Limited, 2022. ISBN 9781638280521. doi: 10.1561/978-1-63828-053-820251002.
- F1000 Research Ltd. Reproducible research data and software. *F1000Research Channels*. doi: 10.12688/f1000research.channels.908.
- Laura Ferrarotti and Alberto Bemporad. Synthesis of optimal feedback controllers from data via stochastic gradient descent. In *2019 18th European Control Conference (ECC)*, pages 2486–2491. IEEE, 2019. doi: 10.23919/ecc.2019.8796130.
- Monique Florenzano and Cuong Le Van. *Convex Optimization With Convex Constraints*. Springer Berlin Heidelberg, 2001. ISBN 9783642625701. doi: 10.1007/978-3-642-56522-9\_7.
- Christodoulos A. Floudas. *Convex Analysis*. Oxford University Press, 1995. ISBN 9780195100563. doi: 10.1093/oso/9780195100563.003.0006.
- Jiwei Fu and Mohammad Foondun. A theoretical and experimental study of gradient descent and its stochastic variants. 2025. doi: 10.36227/techrxiv.176162233.33076025/v1.
- Christopher Gandrud. *Reproducible Research with R and R Studio*. Chapman and Hall/CRC, 2013. ISBN 9781466572850. doi: 10.1201/b15100.
- Christopher Gandrud. *Conclusion*. Chapman and Hall/CRC, 2018a. ISBN 9781315382548. doi: 10.1201/9781315382548-14.
- Christopher Gandrud. *Getting Started with Reproducible Research*. Chapman and Hall/CRC, 2018b. ISBN 9781315382548. doi: 10.1201/9781315382548-2.
- Christopher Gandrud. *Introducing Reproducible Research*. Chapman and Hall/CRC, 2018c. ISBN 9781315382548. doi: 10.1201/9781315382548-1.
- Christopher Gandrud. *Reproducible Research with R and RStudio*. Chapman and Hall/CRC, 2018d. ISBN 9781315382548. doi: 10.1201/9781315382548.
- Christopher Gandrud. *Conclusion*. Chapman and Hall/CRC, 2020a. ISBN 9780429031854. doi: 10.1201/9780429031854-17.
- Christopher Gandrud. *Getting Started with Reproducible Research*. Chapman and Hall/CRC, 2020b. ISBN 9780429031854. doi: 10.1201/9780429031854-3.
- Christopher Gandrud. *Introducing Reproducible Research*. Chapman and Hall/CRC, 2020c. ISBN 9780429031854. doi: 10.1201/9780429031854-2.
- Christopher Gandrud. *Reproducible Research with R and RStudio*. Chapman and Hall/CRC, 2020d. ISBN 9780429031854. doi: 10.1201/9780429031854.

- Sidney Gavel. *Ensuring Your Quantitative Research is Replicable and Reproducible*. SAGE Publications Ltd, 2025. ISBN 9781036216696. doi: 10.4135/9781036216696.
- Eline Van Geert and Lisa Koßmann. Reproducible research reports with quarto. In *proceedings 2025*. PubPub. doi: 10.21428/1192f2f8.fb1a1b7f.
- Ping Guo. Variable selection of regularized stochastic gradient descent in logistic regression. *SCIREA Journal of Mathematics*, 2022. doi: 10.54647/mathematics11319.
- Maja Gwózdź. Stochastic and semi-stochastic gradient descent in speech disambiguation. 2018. doi: 10.7490/f1000research.1115776.1.
- Sajad Fathi Hafshejani, Daya Gaur, Shahadat Hossain, and Robert Benkoczi. Fast armijo line search for stochastic gradient descent. 2023. doi: 10.21203/rs.3.rs-2285238/v1.
- Nir Halman. Approximating convex functions via non-convex oracles under the relative noise model. *Discrete Optimization*, 16:1–16, 2015. doi: 10.1016/j.disopt.2014.12.001.
- Janna Hastings. *AI for Reproducible Research*. CRC Press, 2023. ISBN 9781003226642. doi: 10.1201/9781003226642-4.
- Andy Hector. *Reproducible Research*. Oxford University Press, 2021. ISBN 9780198798170. doi: 10.1093/os/o/9780198798170.003.0004.
- Soroor Hediye-Zadeh and Melissa J. Davis. Computational workflows for research students: towards a reproducible research. 2016. doi: 10.7490/f1000research.1113332.1.
- Thomas F Heston. Statistics, ethics, and the promotion of reproducible research. 2023. doi: 10.22541/au.169627960.06669688/v1.
- Konrad Hinsén. Unifying version control and dependency management for reproducible research. 2012. doi: 10.59350/986d5-0he30.
- Konrad Hinsén. Platforms for reproducible research. 2013a. doi: 10.59350/s56jr-3cn95.
- Konrad Hinsén. Python as a platform for reproducible research. 2013b. doi: 10.59350/85pnj-bak53.
- Konrad Hinsén. From reproducible to verifiable computer-aided research. 2016. doi: 10.59350/fj0g3-5yv59.
- Konrad Hinsén. Reproducible research in the python ecosystem: a reality check. 2017a. doi: 10.59350/wgeb-68y68.
- Konrad Hinsén. Sustainable software and reproducible research: dealing with software collapse. 2017b. doi: 10.59350/7tavk-2hf75.
- Holger Hoefling and Anthony Rossini. *Reproducible Research for Large-Scale Data Analysis*. Chapman and Hall/CRC, 2018. ISBN 9781315373461. doi: 10.1201/9781315373461-8.
- Iain Hrynaszkiewicz, Peter Li, and Scott Edmunds. *Open Science and the Role of Publishers in Reproducible Research*. Chapman and Hall/CRC, 2018. ISBN 9781315373461. doi: 10.1201/9781315373461-15.
- Han Hu. Adaptive batch size time evolving stochastic gradient descent for federated learning\_supp1-3610169.pdf. doi: 10.1109/tpami.2025.3610169/mm1.
- Jie Hu, Vishwaraj Doshi, and Do-Young Eun. Efficiency ordering of stochastic gradient descent. In *Advances in Neural Information Processing Systems 35*, pages 15875–15888. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2022. doi: 10.52202/068431-1155.
- Yuchao Hua, Lingai Luo, Steven Le Corre, and Yilin Fan. Machine-learning topology optimization with stochastic gradient descent optimizer for heat conduction problems. 2023. doi: 10.2139/ssrn.4594476.
- Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993. doi: 10.1016/0925-2312(93)90006-o.

- ITMO University. Testing of the stochastic parallel gradient descent algorithm to the alignment of a two-mirror telescope. , 2020. doi: 10.17586/1023-5086-2020-87-05-31-41.
- Haruki Ito and Yosuke Onoue. Constrained graph drawing by stochastic gradient descent. In *2026 IEEE 19th Pacific Visualization Conference (PacificVis)*, pages 11–21. IEEE, 2026. doi: 10.1109/pacificvis6879.1.2026.00006.
- Franck Iutzeler, Edouard Pauwels, and Samuel Vaiter. Derivatives of stochastic gradient descent in parametric optimization. In *Advances in Neural Information Processing Systems 37*, pages 118859–118882. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-3775.
- Stephen E. Jacobsen. *BB algorithm; Concave programming; D.C. programming; Quadratic knapsack. Quadratic programming with bound constraints; Reverse convex optimization Standard quadratic optimization problems: Theory REVERSE CONVEX OPTIMIZATION*. Springer US, 2001. ISBN 9780792369325. doi: 10.1007/0-306-48332-7\_431.
- Stephen E. Jacobsen. *Reverse Convex Optimization*. Springer US, 2008. ISBN 9780387747583. doi: 10.1007/978-0-387-74759-0\_564.
- Adit Jain and Vikram Krishnamurthy. Controlling stochastic gradient descent using stochastic approximation for robust distributed optimization. *Numerical Algebra, Control and Optimization*, 15(1):173–195, 2025. doi: 10.3934/naco.2024041.
- Zhe Jiao and Martin Keller-Ressel. Emergence of heavy tails in homogenized stochastic gradient descent. In *Advances in Neural Information Processing Systems 37*, pages 14066–14092. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-0450.
- Peter Kedron, Joseph Holler, and Sarah Bardin. Reproducible research practices and barriers to reproducible research in geography: Insights from a survey. 2023. doi: 10.31219/osf.io/nyrq9.
- Nikhil Ketkar. *Stochastic Gradient Descent*. Apress, 2017. ISBN 9781484227657. doi: 10.1007/978-1-4842-2766-4\_8.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. URL <https://arxiv.org/abs/1412.6980>.
- Justin Kitzes. *The Basic Reproducible Workflow Template*. University of California Press, 2019. ISBN 9780520967779. doi: 10.1525/9780520967779-005.
- Tomer Koren, Roi Livni, Yishay Mansour, and Uri Sherman. Benign underfitting of stochastic gradient descent. In *Advances in Neural Information Processing Systems 35*, pages 19605–19617. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2022. doi: 10.52202/068431-1425.
- Marcin Kosinski. coxphsgd: Stochastic gradient descent log-likelihood estimation in cox proportional hazards model. *CRAN: Contributed Packages*, 2017. doi: 10.32614/cran.package.coxphsgd.
- Makinder Kour, surbhi gupta, and SHUBHAM MAHAJAN. Stochastic gradient descent optimized efficient transfer learning architecture for brain tumor segmentation. 2024. doi: 10.2139/ssrn.5012382.
- Aggeliki Koutsibella and Konstantinos D. Koutroumbas. Stochastic gradient descent possibilistic clustering. In *11th Hellenic Conference on Artificial Intelligence*, pages 189–194. ACM, 2020. doi: 10.1145/3411408.3411436.
- Dominic Donald Kramer. *Basis identification through convex optimization*. PhD thesis.
- Yoshihiko Kunisato. Introduction to reproducible psychological research. 2019. doi: 10.31234/osf.io/x8js5.
- Henry Lam and Zitong Wang. Resampling stochastic gradient descent cheaply. In *2023 Winter Simulation Conference (WSC)*, pages 3681–3692. IEEE, 2023. doi: 10.1109/wsc60868.2023.10408023.
- Guanghui Lan. *Stochastic Gradient Descent*. Springer International Publishing, 2023. ISBN 9783030546212. doi: 10.1007/978-3-030-54621-2\_777-1.

- Jean B. Lasserre. On convex optimization without convex representation. *Optimization Letters*, 5(4):549–556, 2011. doi: 10.1007/s11590-011-0323-1.
- Jean B. Lasserre. Erratum to: On convex optimization without convex representation. *Optimization Letters*, 8(5):1795–1796, 2014. doi: 10.1007/s11590-014-0735-9.
- Brandon C LeBeau, Ariel M Aloe, and Scott Ellison. Reproducible analyses in educational research. 2020. doi: 10.17077/pp.005637.
- Jaewoo Lee. Differentially private variance reduced stochastic gradient descent. In *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, pages 161–166. IEEE, 2017. doi: 10.1109/ictcs.2017.60.
- Chris Junchi Li. Withdrawn: Bridging stochastic gradient descent and markov chains: Constant step-size convergence and richardson-romberg extrapolation. 2024a. doi: 10.21203/rs.3.rs-5202271/v1.
- Chris Junchi Li. Withdrawn: Bridging stochastic gradient descent and markov chains: Constant step-size convergence and richardson-romberg extrapolation. 2024b. doi: 10.21203/rs.3.rs-5202271/v2.
- Dequan Li, Yuheng Zhang, and Yuejin Zhou. Fast distributed stochastic nesterov gradient descent algorithm for image classification. In *2021 China Automation Congress (CAC)*, pages 6408–6413. IEEE, 2021. doi: 10.1109/cac53003.2021.9727635.
- Junchi Li. Stochastic gradient descent in nonconvex optimization: Continuous-time dynamics and the role of learning rates. 2024c. doi: 10.2139/ssrn.4980990.
- Li Li. *Convex Relaxation*. Springer Berlin Heidelberg, 2015. ISBN 9783662463550. doi: 10.1007/978-3-662-46356-7\_6.
- Nicolas Limare. *Reproducible research, software quality, online interfaces and publishing for image processing*. PhD thesis.
- Dongyang Liu, Zeqiang Chen, and Nengcheng Chen. Optimizing synchronous stochastic gradient descent with local efficient sign and model averaging correction. 2024. doi: 10.2139/ssrn.4965637.
- Tianyi Liu, Zhehui Chen, Enlu Zhou, and Tuo Zhao. A diffusion approximation theory of momentum stochastic gradient descent in nonconvex optimization. *Stochastic Systems*, 11(4):307–323, 2021. doi: 10.1287/stsy.2021.0083.
- Xinfu Liu. *Autonomous Trajectory Planning by Convex Optimization*. PhD thesis.
- Roi Livni. The sample complexity of gradient descent in stochastic convex optimization. In *Advances in Neural Information Processing Systems 37*, pages 64215–64241. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-2048.
- Seraphine Maerz. Quarto in rstudio: Writing reproducible & dynamic research papers. Technical report, 2024.
- Ben Marwick. *Case Study 12: Using R and Related Tools for Reproducible Research in Archaeology*. University of California Press, 2019. ISBN 9780520967779. doi: 10.1525/9780520967779-021.
- Pierre Maréchal. *Generating Convex Functions*. Springer US, 2001. ISBN 9780792369424. doi: 10.1007/978-1-4613-0279-7\_21.
- Jacob Mattingley and Stephen Boyd. *Automatic code generation for real-time convex optimization*. Cambridge University Press, 2009. doi: 10.1017/cbo9780511804458.002.
- MISSING-VALUE MISSING-VALUE. Platforms. *Implementing Reproducible Research*, pages 361–362, a. doi: 10.1201/b16868-20.
- MISSING-VALUE MISSING-VALUE. Practices and guidelines. *Implementing Reproducible Research*, pages 167–168, b. doi: 10.1201/b16868-12.

- MISSING-VALUE MISSING-VALUE. Tools. *Implementing Reproducible Research*, pages 21–22, c. doi: 10.1201/b16868-6.
- Akshay Mittal. Explainable ai-augmented devsecops for secure and reproducible cloud-native research software. 2025. doi: 10.36227/techrxiv.175617187.78775647/v1.
- Louis Moresi. Alaska moho model (reproducible research with containers). 2018a. doi: 10.59350/pn8gh-98592.
- Louis Moresi. Alaska moho model (reproducible research with containers). 2018b. doi: 10.59350/15qd1-96r10.
- Samrat Mukhopadhyay. Stochastic gradient descent for linear systems with sequential matrix entry accumulation. *Signal Processing*, 171:107494, 2020. doi: 10.1016/j.sigpro.2020.107494.
- Kazuo Murota. *L-Convex Functions and M-Convex Functions*. Springer US, 2001. ISBN 9780792369325. doi: 10.1007/0-306-48332-7\_244.
- Kazuo Murota. *L-convex Functions and M-convex Functions*. Springer US, 2008. ISBN 9780387747583. doi: 10.1007/978-0-387-74759-0\_325.
- Kazuo Murota. *L-Convex Functions and M-Convex Functions*. Springer Nature Switzerland, 2024. ISBN 9783030546212. doi: 10.1007/978-3-030-54621-2\_325-1.
- Peter Murray-Rust and Dave Murray-Rust. *Reproducible Physical Science and the Declaratron*. Chapman and Hall/CRC, 2018. ISBN 9781315373461. doi: 10.1201/9781315373461-5.
- Yurii Nesterov. *Nonsmooth Convex Optimization*. Springer US, 2004a. ISBN 9781461346913. doi: 10.1007/978-1-4419-8853-9\_3.
- Yurii Nesterov. *Smooth Convex Optimization*. Springer US, 2004b. ISBN 9781461346913. doi: 10.1007/978-1-4419-8853-9\_2.
- Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1): 125–161, 2013. doi: 10.1007/s10107-012-0629-5.
- Yurii Nesterov. *Nonsmooth Convex Optimization*. Springer International Publishing, 2018a. ISBN 9783319915777. doi: 10.1007/978-3-319-91578-4\_3.
- Yurii Nesterov. *Smooth Convex Optimization*. Springer International Publishing, 2018b. ISBN 9783319915777. doi: 10.1007/978-3-319-91578-4\_2.
- Thanh Huy Nguyen. *Heavy-tailed nature of stochastic gradient descent in deep learning : theoretical and empirical analysis*. PhD thesis.
- Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Science & Business Media, 2nd edition, 2006. ISBN 978-0-387-30303-1. doi: 10.1007/978-0-387-40065-5.
- Tazro Ohta and Osamu Ogasawara. Container-based sequence data analysis workflow for reproducible research. 2015. doi: 10.7490/f1000research.1110170.1.
- Figen S. Oktem, Liang Gao, and Farzad Kamalabadi. *Computational Spectral and Ultrafast Imaging via Convex Optimization*. Springer International Publishing, 2017. ISBN 9783319616087. doi: 10.1007/978-3-319-61609-4\_5.
- PeerJ. Algorithm 1 : Convex optimization with regularized feature selection. a. doi: 10.7717/peerj-cs.3752/table-101.
- PeerJ. Figure 2: The final reproducible research criteria used for the evaluation. b. doi: 10.7717/peerj.5072/fig-2.
- PeerJ. Supplemental information 1: Reproducible research instructions. c. doi: 10.7717/peerj-cs.904/supp-1.

- PeerJ. Table 4: Stochastic gradient descent (sgd) parameter settings with cc images. d. doi: 10.7717/peerj-cs.3332/table-4.
- Roger D Peng. Reproducible research in computational science. *Science*, 334(6060):1226–1227, 2011. doi: 10.1126/science.1213847.
- Paul Peseux, Thierry Paquet, and Maxime Berar. Stochastic gradient descent with gradient estimator for categorical features. 2023. doi: 10.2139/ssrn.4439301.
- Juan Peypouquet. *Convex Analysis and Subdifferential Calculus*. Springer International Publishing, 2015. ISBN 9783319137094. doi: 10.1007/978-3-319-13710-0\_3.
- Adrien Peyrache. elife assessment: Spyglass: a framework for reproducible and shareable neuroscience research. 2025. doi: 10.7554/elife.108089.1.sa2.
- Adrien Peyrache. elife assessment: Spyglass: a framework for reproducible and shareable neuroscience research. 2026. doi: 10.7554/elife.108089.2.sa3.
- Loucas Pillaud-Vivien. *Learning with reproducing kernel Hilbert spaces : stochastic gradient descent and laplacian estimation*. PhD thesis.
- Russell Poldrack. *Case Study 29: Developing a Reproducible Workflow for Large-Scale Phenotyping*. University of California Press, 2019. ISBN 9780520967779. doi: 10.1525/9780520967779-038.
- Likit Preeyanon, Alexis Black Pyrkosz, and C. Titus Brown. *Reproducible Bioinformatics Research for Biologists*. Chapman and Hall/CRC, 2018. ISBN 9781315373461. doi: 10.1201/9781315373461-7.
- Princeton University Press. *Appendix:.* Princeton University Press, 2020a. ISBN 9780691200316. doi: 10.2307/j.ctvqsdxd.15.
- Princeton University Press. *Appendix: Executive Summary on Efficient Solvability of Convex Optimization Problems*. Princeton University Press, 2020b. ISBN 9780691200316. doi: 10.1515/9780691200316-013.
- PubPub. Ninbioinformatics reproducible research reports. doi: 10.21428/3c290898.
- Fabio Ramos and Lionel Ott. Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent. In *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, 2015. doi: 10.15607/rss.2015.xi.002.
- M. Ravasi, T. Selvan Pandurangan, and N. Luiken. Multi-dimensional deconvolution with stochastic gradient descent. In *83rd EAGE Annual Conference & Exhibition*, pages 1–5. European Association of Geoscientists & Engineers, 2022. doi: 10.3997/2214-4609.202210234.
- Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *ICLR*, 2018. URL <https://arxiv.org/abs/1904.09237>.
- Rok Roškar. Renku: a platform for collaborative, reusable and reproducible research. 2022. doi: 10.5194/egusphere-egu22-10697.
- Michael Rübsamen, Amr El-Keyi, Alex B. Gershman, and Thia Kirubarajan. *Robust broadband adaptive beamforming using convex optimization*. Cambridge University Press, 2009. doi: 10.1017/cbo9780511804458.010.
- SAGE Publications Ltd. Transparent and reproducible data analysis. *SAGE Research Methods Foundations*, 2020. doi: 10.4135/9781526421036926480.
- Sage Publications, Ltd. Preparing for transparent and reproducible quantitative social science research. 2025. doi: 10.4135/9781036230647.
- N.N. Schraudolph. Local gain adaptation in stochastic gradient descent. In *9th International Conference on Artificial Neural Networks: ICANN '99*, volume 1999, pages 569–574. IEE, 1999. doi: 10.1049/cp:19991170.
- Anuraganand Sharma. Guided stochastic gradient descent algorithm for inconsistent datasets. *Applied Soft Computing*, 73:1068–1080, 2018. doi: 10.1016/j.asoc.2018.09.038.

- Anuraganand Sharma. Guided parallelized stochastic gradient descent for delay compensation. *Applied Soft Computing*, 102:107084, 2021. doi: 10.1016/j.asoc.2021.107084.
- Sapna Shrimali, Govind S. Sharma, and Sunil K. Srivastava. A comparative study of gradient descent and stochastic gradient descent method for optimization. In *AIP Conference Proceedings*, volume 2768, page 020002. AIP Publishing, 2023. doi: 10.1063/5.0148736.
- I. Singer. *On Duality for Quasi-convex Supremization and Reverse Convex Infimization*. Springer US, 2001. ISBN 9781402000096. doi: 10.1007/978-1-4613-0295-7\_16.
- Ivan. Singer. Duality in quasi-convex supremization and reverse convex infimization via abstract convex analysis, and applications to approximation \*\*. *Optimization*, 45(1-4):255–307, 1999. doi: 10.1080/02331939908844436.
- Justin Sirignano and Konstantinos Spiliopoulos. Stochastic gradient descent in continuous time. *SSRN Electronic Journal*, 2017. doi: 10.2139/ssrn.2954149.
- Justin Sirignano and Konstantinos Spiliopoulos. Stochastic gradient descent in continuous time: A central limit theorem. *Stochastic Systems*, 10(2):124–151, 2020. doi: 10.1287/stsy.2019.0050.
- Frederick Solt and Kellen Gracey. icpsrdata: Reproducible data retrieval from the icpsr archive. *CRAN: Contributed Packages*, 2016. doi: 10.32614/cran.package.icpsrdata.
- Frederick Solt and Yue Hu. pewdata: Reproducible retrieval of pew research center datasets. *CRAN: Contributed Packages*, 2016. doi: 10.32614/cran.package.pewdata.
- Chongya Song. *An Angle-based Stochastic Gradient Descent Method for Machine Learning: Principle and Application*. PhD thesis.
- Chongya Song, Alexander Pons, and Kang Yen. Ag-sgd: Angle-based stochastic gradient descent. *IEEE Access*, 9:23007–23024, 2021. doi: 10.1109/access.2021.3055993.
- SPIE-Intl Soc Optical Eng. 10.1117/12.2305101.5783306294001. a. doi: 10.1117/12.2305101.5783306294001.
- SPIE-Intl Soc Optical Eng. 10.1117/12.2512817.6013939792001. b. doi: 10.1117/12.2512817.6013939792001.
- SPIE-Intl Soc Optical Eng. 10.1117/12.2525953.6062680904001. c. doi: 10.1117/12.2525953.6062680904001.
- SPIE-Intl Soc Optical Eng. 10.1117/12.2599744.6269411057001. d. doi: 10.1117/12.2599744.6269411057001.
- SPIE-Intl Soc Optical Eng. 10.1117/12.2643564.6314793833112. e. doi: 10.1117/12.2643564.6314793833112.
- SPIE-Intl Soc Optical Eng. 10.1117/12.3026153.0df877b5-f9b8-ee11-a99d-c49c781f4d15. f. doi: 10.1117/12.3026153.0df877b5-f9b8-ee11-a99d-c49c781f4d15.
- SPIE-Intl Soc Optical Eng. 10.1117/12.3028105.e628af96-d7c5-ee11-a99e-00505691c5e1. g. doi: 10.1117/12.3028105.e628af96-d7c5-ee11-a99e-00505691c5e1.
- {Springer Science and Business Media LLC}. Fostering reproducible fmri research. *Nature Communications*, 8(1), 2017. doi: 10.1038/ncomms14748.
- Springer-Verlag. Convex discrete optimization. *SpringerReference*, a. doi: 10.1007/springerreference\_72172.
- Springer-Verlag. Convex envelopes in optimization problems. *SpringerReference*, b. doi: 10.1007/springerreference\_72173.
- Springer-Verlag. Duality theory: Monoduality in convex optimization. *SpringerReference*, c. doi: 10.1007/springerreference\_72219.
- Springer-Verlag. Global optimization: Tight convex underestimators. *SpringerReference*, d. doi: 10.1007/springerreference\_72325.
- Springer-Verlag. Reverse convex optimization, reverse convex programming. *SpringerReference*, e. doi: 10.1007/springerreference\_72642.

- Ashwin Kumar Srinivasan. An intelligent predictive cpu scheduling algorithm using online multiple linear regression and stochastic gradient descent. 2026. doi: 10.2139/ssrn.6956552.
- John Stillwell. Statistical inference via convex optimization. 2019. doi: 10.23943/princeton/9780691197296.001.0001.
- Victoria Stodden. *Implementing Reproducible Research*. Chapman and Hall/CRC, 2014. ISBN 9781466561601. doi: 10.1201/b16868.
- Julia Feld Strand and Violet Aurora Brown. Publishing open, reproducible research with undergraduates. 2019. doi: 10.31234/osf.io/f7kuy.
- Roman G. Strongin and Yaroslav D. Sergeyev. *Global Optimization under Non-Convex Constraints — The Index Approach*. Springer US, 2000. ISBN 9781461371175. doi: 10.1007/978-1-4615-4677-1\_6.
- Peter Suber. Oa for reproducible research. 2008a. doi: 10.63485/4jjac-1dd97.
- Peter Suber. Open licensing to enable reproducible research. 2008b. doi: 10.63485/8k73p-vrh09.
- Peter Suber. More on oa to facilitate reproducible research. 2009. doi: 10.63485/tys9t-3y862.
- Changho Suh. *1. Convex Optimization Basics*. Now Publishers, 2022. ISBN 9781638280521. doi: 10.1561/9781638280538.ch1.
- Hezhe Sun, Nachuan Yang, and Yuzhe Li. Noisy stochastic gradient descent algorithm with stochastic event-triggered mechanism for communication-efficient distributed learning. 2023. doi: 10.2139/ssrn.4588764.
- Tao Sun. On the decentralized stochastic gradient descent with markov chain sampling\_supp1-3297053.pdf. doi: 10.1109/tsp.2023.3297053/mm1.
- Ramesh T and Santhi V. Hybrid optimization model integrating gradient descent and stochastic descent for enhanced osteoporosis and osteopenia recognition. *Journal of Machine and Computing*, pages 340–348, 2024. doi: 10.53759/7669/jmc202404032.
- Aliasghar Tarkhan and Noah Simon. bigsurvsd: Big survival analysis using stochastic gradient descent. *CRAN: Contributed Packages*, 2020. doi: 10.32614/cran.package.bigsurvsd.
- Sergios Theodoridis. *Stochastic Gradient Descent*. Elsevier, 2015. ISBN 9780128015223. doi: 10.1016/b978-0-12-801522-3.00005-7.
- Sergios Theodoridis. *Online Learning: the Stochastic Gradient Descent Family of Algorithms*. Elsevier, 2020. ISBN 9780128188033. doi: 10.1016/b978-0-12-818803-3.00014-3.
- Sergios Theodoridis. *Online learning: the stochastic gradient descent family of algorithms*. Elsevier, 2026. ISBN 9780443292385. doi: 10.1016/b978-0-44-329238-5.00011-1.
- Mehmet Yigit Turali and Suleyman Serdar Kozat. Optimal stochastic gradient descent algorithm for filtering. 2024. doi: 10.2139/ssrn.4879640.
- Daniel Turek and Fatma Deniz. *Case Studies in Reproducible Research*. University of California Press, 2019. ISBN 9780520967779. doi: 10.1525/9780520967779-006.
- Stephen Turner. Seminar: Reproducible research with r, latex, & amp; sweave. 2009a. doi: 10.59350/2958f-vdz74.
- Stephen Turner. Seminar: Reproducible research with r, latex, & amp; sweave. 2009b. doi: 10.59350/d3cd0-5jy93.
- Stephen Turner. Sweave for reproducible research and beautiful statistical reports. 2010a. doi: 10.59350/247k5-kfv86.
- Stephen Turner. Sweave for reproducible research and beautiful statistical reports. 2010b. doi: 10.59350/pexqx-j5g86.

- Hoang Tuy. *Partly Convex and Convex-Monotonic Optimization Problems*. Springer-Verlag. ISBN 3540230270. doi: 10.1007/3-540-27170-8\_37.
- Hoang Tuy. *Convex Functions*. Springer US, 1998a. ISBN 9781441947833. doi: 10.1007/978-1-4757-2809-5\_2.
- Hoang Tuy. *Convex Sets*. Springer US, 1998b. ISBN 9781441947833. doi: 10.1007/978-1-4757-2809-5\_1.
- Hoang Tuy. *Convex Functions*. Springer International Publishing, 2016a. ISBN 9783319314822. doi: 10.1007/978-3-319-31484-6\_2.
- Hoang Tuy. *Convex Sets*. Springer International Publishing, 2016b. ISBN 9783319314822. doi: 10.1007/978-3-319-31484-6\_1.
- University of California Press. *ELEVEN. Reproducible Workflow*. University of California Press, 2019a. doi: 10.1525/9780520969230-014.
- University of California Press. *Introduction*. University of California Press, 2019b. doi: 10.2307/j.ctvpb3xkg.6.
- University of California Press. *Reproducible Workflow*. University of California Press, 2019c. doi: 10.2307/j.ctvpb3xkg.16.
- University of California Press. *Tables*. University of California Press, 2019d. doi: 10.1525/9780520969230-002.
- University of California Press. *What Is Ethical Research?* University of California Press, 2019e. doi: 10.2307/j.ctvpb3xkg.7.
- David Wang and Andrew Murphy. Stochastic gradient descent. *Radiopaedia.org*, 2018. doi: 10.53347/rid-61715.
- John Wang and Edwin Olson. Robust pose graph optimization using stochastic gradient descent. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4284–4289. IEEE, 2014. doi: 10.1109/icra.2014.6907482.
- Eric Weisbrod. example-project: A reproducible empirical research template. 2026. doi: 10.31235/osf.io/yx7af\_v1.
- Ethan P. White. Software skills for reproducible data-intensive research. 2018. doi: 10.7490/f1000research.1115901.1.
- Kristina Wiebels and David Moreau. Leveraging containers for reproducible psychological research. 2021. doi: 10.31234/osf.io/h7tkg.
- R.G.J. Wijnhoven and P.H.N. de With. Fast training of object detection using stochastic gradient descent. In *2010 20th International Conference on Pattern Recognition*, pages 424–427. IEEE, 2010. doi: 10.1109/icpr.2010.112.
- Wiley. Convex functions. *Convex Optimization*, pages 67–109, 2021a. doi: 10.1002/9781119804093.ch3.
- Wiley. Convex sets. *Convex Optimization*, pages 29–66, 2021b. doi: 10.1002/9781119804093.ch2.
- Wiley. Generalizations of convex functions. *Convex Optimization*, pages 111–136, 2021c. doi: 10.1002/9781119804093.ch4.
- Austin Williams, Noah Walton, Austin Maryanski, Sandra Bogetic, Wes Hines, and Vladimir Sobes. Stochastic gradient descent for optimization of nuclear systems. 2022. doi: 10.21203/rs.3.rs-2073277/v1.
- Yihui Xie. *knitr: A Comprehensive Tool for Reproducible Research in R*. Chapman and Hall/CRC, 2018. ISBN 9781315373461. doi: 10.1201/9781315373461-1.

- Shakiba Yaghoubi and Georgios Fainekos. Hybrid approximate gradient and stochastic descent for falsification of nonlinear systems. In *2017 American Control Conference (ACC)*, pages 529–534. IEEE, 2017. doi: 10.23919/acc.2017.7963007.
- Kazunori D Yamada. Hyperparameter-free optimizer of stochastic gradient descent that incorporates unit correction and moment estimation. 2018. doi: 10.1101/348557.
- X. M. Yang. On e-convex sets, e-convex functions, and e-convex programming. *Journal of Optimization Theory and Applications*, 109(3):699–704, 2001. doi: 10.1023/a:1017532225395.
- Zhuang Yang and Li Ma. Adaptive stochastic gradient descent for large-scale learning problems. 2022. doi: 10.21203/rs.3.rs-1066512/v1.
- Anil Yildiz and Julia Kowalski. Data-integrated executable publications for reproducible geohazards research. 2023. doi: 10.5194/egusphere-egu23-7417.
- E. A. Youness. E-convex sets, e-convex functions, and e-convex programming. *Journal of Optimization Theory and Applications*, 102(2):439–450, 1999. doi: 10.1023/a:1021792726715.
- Erik Zamora and Humberto Sossa. Dendrite morphological neurons trained by stochastic gradient descent. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016. doi: 10.1109/ssci.2016.7849933.
- Alexander J. Zaslavski. *Minimization of Sharp Weakly Convex Functions*. Springer International Publishing, 2020a. ISBN 9783030378219. doi: 10.1007/978-3-030-37822-6\_11.
- Alexander J. Zaslavski. *Nonsmooth Convex Optimization*. Springer International Publishing, 2020b. ISBN 9783030602994. doi: 10.1007/978-3-030-60300-7\_2.
- Alexander J. Zaslavski. *PDA-Based Method for Convex Optimization*. Springer International Publishing, 2020c. ISBN 9783030378219. doi: 10.1007/978-3-030-37822-6\_9.
- Constantin Zălinescu. On locally lipschitz convex functions defined on subsets with empty interior of locally convex spaces. *Optimization*, pages 1–21, 2024. doi: 10.1080/02331934.2024.2388200.