Optimized Control of Duplex Networks

Haoyu Zheng, Xizhe Zhang

Abstract—Many real-world complex systems can be modeled as *multiplex* networks, where each layer represents a distinct set of interactions among the same entities. Controlling such systems, defined as steering the system to desired states with external inputs, is crucial in various domains. However, existing network control theory largely focuses on single-layer networks, and applying separate controls to each layer of a multiplex system often yields a redundant set of driver nodes, increasing cost and complexity. To address this gap, we formulate the Universal Minimum Union Driver Set (MinUDS) problem for duplex networks, which seeks the smallest union of layer-wise driver nodes (i.e., the fewest unique nodes to be actuated) that renders both layers structurally controllable under fixed driver budgets. We propose a novel algorithm, *Shortest Cross-Layer Augmenting Path Search* (CLAP-S), that efficiently navigates the combinatorial search space of control configurations. By introducing *Cross-Layer Augmenting Path* (CLAP), CLAP-S iteratively realigns each layer's MDS to maximize their overlap. We prove the algorithm's global optimality and demonstrate its efficiency on both synthetic networks and real-world multiplex systems. The results show that CLAP-S consistently outperforms baseline approaches by reducing the number of required driver nodes and computational time by an order of magnitude. This work provides a powerful, general-purpose tool for optimizing control strategies in multi-layer networks, enabling more economical interventions in diverse fields. *Code and data.* Our implementation of CLAP-S and all baselines, together with scripts to reproduce the figures and tables, is available at https://github.com/njnklab/CLAP-S_Algorithm.

Index Terms—Multiplex Network; Structural Controllability; Driver Nodes; Network Control; Maximum Matching

1 Introduction

The study of complex systems, from the intricate web of molecular interactions within a cell to the vast architecture of the internet, is a central theme in modern science [1], [2]. Network science provides a powerful mathematical framework for modeling these systems, where entities are represented as nodes and their interactions as edges. A fundamental challenge in this domain is network control: the ability to steer a system's dynamics from any initial state to any desired final state through external inputs applied to a subset of its nodes [3]. Achieving this goal with minimal intervention is critical for applications ranging from designing therapeutic strategies in systems biology to implementing efficient interventions in social networks [4], [5].

The modern era of network control began with the structural controllability framework, which determines a system's controllability based solely on its network topology [6]. A seminal contribution by Liu et al. [5] demonstrated that identifying the **Minimum Driver Set (MDS)** required for structural control is equivalent to solving a maximum matching problem on the network's corresponding bipartite graph. This elegant mapping rendered the problem computationally tractable and spurred a wave of research [7], [8]. This line of work also clarified inherent hardness and design tradeoffs (e.g., minimal controllability is NP-hard and calls for structural I/O frameworks), which shaped subsequent algorithmic directions [9], [10]. Subsequent work in single-layer networks has focused on refining this concept, developing algorithms to find all possible MDS configurations [11], defining metrics like control capacity to quantify a node's importance across these configurations [12], [13], and exploring diverse control strategies, such as altering indispensable proteins [14], targeting specific communities [15], adapting to dynamic networks [16], [17], or modifying the network's edge structure [18], [19], [20]. These theoretical advances have been paralleled by impactful applications, from identifying cancer-keeper genes as therapeutic targets [21] and guiding the design of brain stimulation therapies [4], [22], to bridging control theory with influence maximization in social networks [23].

However, most real-world systems are inherently multilayer, where the same entities interact through multiple modalities (e.g., anatomical vs. functional; regulation vs. interaction) [24], [25], [26]. The structural coupling among layers can create behaviors absent in isolated layers, as extensively documented in interdependent networks and robustness studies [27], [28]. A cell operates through interconnected layers of gene regulation and protein interactions [29]; the brain comprises layers of structural and functional connectivity [4]; and modern society is woven from multiplex social and communication networks [30]. Applying control theory to such systems introduces significant new challenges. A naive approach of finding an MDS for each layer independently and then taking their union is conceptually simple but practically inefficient, often leading to a redundant and costly set of driver nodes. Some multilayer studies adopt a stringent placement rule that enforces identical driver locations across layers [31], which can be overly restrictive in practice; here we instead aim to minimize the cross-layer union of drivers while preserving layer-wise minimality.

For multilayer systems, several lines of work analyze how layering changes controllability and input placement. A central thread extends the maximum-matching frame-

- *Early Intervention Unit, Department of Psychiatry, Nanjing Brain Hospital, Nanjing Medical University, Nanjing 210029, China.*
- *School of Biomedical Engineering and Informatics, Nanjing Medical University, Nanjing 211166, China.*
- *Corresponding author: Xizhe Zhang (zhangxizhe@njmu.edu.cn).*

work to multiplex settings and shows that cross-layer structure can fundamentally reshape driver requirements, sometimes stabilizing configurations that are not available in separate layers [31]. Another thread studies two-layer (duplex) networks with different timescales, proving that which layer receives inputs dramatically alters the minimum input count—favoring the faster layer under separation [32]. Beyond full-state controllability, recent studies explore target controllability and heterogeneous high-dimensional nodes in multilayered systems, emphasizing how inter-layer couplings can render a system controllable even when an individual layer is not [33], [34], [35], [36]. There is also work constraining all drivers to a single designated layer or otherwise tying input locations to one layer, which addresses practical placement limits but differs from our objective of jointly optimizing across layers [37], [38]. Complementary strands investigate energy and spectral alignment across layers or spreading-process control, again highlighting structural interactions across layers rather than layer-wise unions of driver sets [39], [40].

A separate body of literature tackles multilayer controllability via alternative graph-theoretic surrogates, most notably minimum dominating sets (MDS-D) and feedback vertex sets (FVS). Dominating-set approaches yield multilayer driver heuristics for undirected or simplified settings [41], whereas FVS-based control targets nonlinear dynamics and attractor steering. Notably, Zheng *et al.* formalized a *minimum union* objective in the nonlinear multilayer setting—minimizing the union of layer-wise FVS driver sets—and proposed greedy procedures for that union minimization [42]. These paradigms optimize different control objectives (domination or cycle breaking) from the matching-based structural controllability we pursue; nevertheless, they motivate our emphasis on minimizing cross-layer redundancy.

Against this landscape, our focus is the *union minimization of matching-based MDS across duplex networks*. Instead of requiring identical drivers across layers [31] or restricting inputs to a single layer [38], we allow each layer to have its own MDS and aim to *reduce the size of their union*. Methodologically, we leverage deterministic, graph-theoretic reconfigurations of layer-wise maximum matchings to increase overlap across layers. Empirically, we compare our approach to random-sampling style baselines that draw multiple maximum matchings per layer and pick the smallest observed union—an approach aligned with prior uses of sampling to probe the space of MDS configurations and node participation frequencies [12]. Our results (Sections 5) show consistent reductions in required drivers and runtime across synthetic and real multiplexes, indicating that targeted cross-layer reconfiguration can substantially outperform undirected sampling strategies.

Our formulation targets settings in which (i) the underlying interactions are intrinsically*directed* and layer-dependent, while (ii) actuation hardware is deployed at shared physical entities and can be reused across layers, making the deployment cost naturally depend on the number of *unique* actuated nodes. Representative examples include condition- or cell-type–specific transcriptional gene regulatory networks, where edges are directed from transcription factors to target genes and reference/inferred directed regulatory graphs are widely studied [43], [44]; multiplex financial networks, where directed exposures/ownership relations among institutions can be decomposed into layers by instrument type or relation semantics [45], [46]; and online information/influence systems, whose follow relationship is asymmetric and thus naturally modeled as a directed graph, coexisting with other directed interaction layers such as mentions/retweets [47]. In such domains, structural controllability is naturally posed on directed graphs [5], and minimizing $|D^{(1)} \cup D^{(2)}|$ captures the number of unique entities that must be instrumented while guaranteeing the controllability of each layer.

This work makes three contributions to the study of structural controllability in duplex networks:

(i) **Graph-theoretic framework for cross-layer controllability.** We formalize cross-layer augmenting paths (CLAP) that trigger budget-preserving driver exchanges across layers, and show that the absence of any CLAP certifies *global optimality* for the union objective (the *CLAP-or-Optimal* result).

(ii) **Deterministic algorithm based on shortest CLAP search.** Our procedure CLAP-S performs a layer-alternating BFS to find a shortest CLAP; each successful exchange strictly reduces Δ by 2 (hence $|U|$ by 1) while preserving (k_1, k_2), and halts with a verifiable optimality certificate when no CLAP exists.

(iii) **Extensive empirical evaluation on synthetic and real multiplexes.** On Erdős–Rényi, Barabási–Albert, and hybrid duplex networks, as well as diverse real-world systems (biological, neuronal, social, and human relationship networks), CLAP-Sachieves *exactly the same* optimal union sizes as an exact integer linear programming (ILP) formulation on every instance, yet runs substantially faster in wall-clock time.

The remainder of the paper is structured as follows. Section 2 reviews background and related work. Section 3 formalizes notation, the duplex model, and the union-minimization problem under per-layer minimality. Section 4 presents the CLAP framework and the main theorems with proof sketches, also details the algorithms (CLAP-S) and computational complexity. Section 5 reports experiments on synthetic and real networks, including ablations and scalability. Finally, Section 6 discusses extensions, limitations, and practical implications.

2 Related Works

Many real systems are *multilayer* (multiplex or interdependent) networks in which the same entities interact through multiple modalities (e.g., anatomical vs. functional brain connectivity, gene–protein–disease relations, cyber–physical infrastructures). Foundational surveys standardized the representation of node–layer tuples, intra-/inter-layer couplings, and layer correlations [24], [25], [26]. Against this backdrop, the seminal structural-controllability result of Liu *et al.* grounded network control in maximum matching: unmatched vertices in a maximum matching of the bipartite representation form a minimum driver-node set (MDS) [5]. Early extensions from single-layer to duplex/multilayer settings clarified *how* cross-layer coupling rules reshape controllability. Menichetti *et al.* mapped duplex controllability

to matching-theoretic objects under a strong coordination assumption—identical driver locations across layers—and revealed phase-like transitions driven by interlayer constraints [31]. Pósfai *et al.* analyzed two-layer systems with disparate time scales and actuation confined to one layer, showing that timescale separation qualitatively alters the driver requirement and which layer should host the inputs [32]. Complementing these, Zhang *et al.* showed in duplex scale-free settings that selecting peripheral interlayer connectors or drivers can enlarge the controllable subspace at a lower cost [37]. Together, these works mark the *origin* of duplex controllability as a problem where layer structure, coupling pattern, and input coordination are inseparable.

Subsequent studies diversified the modeling assumptions and algorithmic targets while keeping multilayer controllability central. One branch translates multilayer control into alternative graph surrogates: Nacher *et al.* formulated multilayer controllability via minimum dominating sets and provided formulas/algorithms for up to six layers [48]. Another branch extracts *common* cross-layer control structures beyond node sets; the "conserved control path" paradigm uses weighted matchings to detect signal-propagation paths that are consistent across layers and demonstrates benefits on multilayer pan-cancer data [49]. Parallel to multilayer modeling, the single-layer literature matured a *matching/alternating-path toolkit* that we leverage as building blocks for duplex coordination: preferential matching exposes degree-structured MDSs [50]; the *input graph* reveals the geometry linking all possible input nodes, with alternating paths certifying mutual substitutability [51]; an efficient procedure enumerates *all* possible input nodes from a single maximum matching via alternating-path searches [11]; and edge-level interventions use alternating/augmenting structures to *alter control modes* while preserving correctness [18], [19]. These results supply principled ways to reconfigure MDSs *within* a layer without losing minimality, suggesting that a duplex coordination mechanism could realign layer-wise MDSs via alternating-path exchanges.

In recent years, several trends have further tightened the connection between multilayer modeling and input *reuse* across "variants." First, *ensemble/uncertainty* viewpoints consider families of realizations and study controllers or architectures that perform across all members [52]; while not explicitly multilayer, they emphasize the economy and commonality of actuation. Second, *energy-aware* placement has clarified how constraining the longest control chain reduces energy and provides implementable guidance on real networks, including human brain connectomes [53], [54]. Third, application-driven multilayer studies have proliferated: in systems biology, multilayer control has been used for target identification in disease networks with an explicit *minimum-union* notion (in a nonlinear-control setting) [55]; in CPS/power systems, structural and strong-structural controllability highlight the role of a small set of generator/communication nodes under uncertainty [56], [57]; in transportation, controllability concepts inform actuator placement, such as ramp meters on multilayer urban-mobility networks [58], [59]; and in neuroengineering, network-control analysis motivates stable stimulation loci across modalities [4], [54]. Across these domains, the engineering desideratum is clear: *reuse* scarce actuators across layers/modalities while keeping each layer minimally actuated.

Gap and positioning. Despite this progress, prior duplex/multilayer controllability works stop short of the following *layer-coordination* objective: *to keep every layer minimally actuated (each layer selects a maximum matching) while minimizing the union of driver nodes across layers*. Independent per-layer selection ignores reuse; enforcing identical drivers across layers [31] maximizes reuse but can be far from optimal when layers differ; minimum-union formulations in biology [55] and common-structure/path approaches [49] optimize different surrogates or operate outside matching-based structural controllability. What is missing is a *matching-theoretic* co-design that (i) searches *within* the space of layer-wise maximum matchings (thereby preserving per-layer minimality) and (ii) *aligns* their unmatched sets to reduce the cross-layer union.

Our perspective. is that we address this gap by coupling layer-wise maximum matchings through cross-layer alternating operations. Conceptually, we lift the well-known alternating-path exchanges from single-layer reconfiguration [11], [18], [19], [51]—grounded in Berge-type optimality and augmenting-path certificates [60], [61] and informed by matching reconfiguration [62]—to a duplex setting. The result is a constructive mechanism that (a) preserves per-layer minimality by staying within maximum matchings, (b) monotonically reduces the union when a cross-layer alternating structure exists, and (c) halts with a verifiable no-augmenting-structure certificate for the union objective. This directly targets the actuator-reuse need documented in brain, biological, CPS/power, transportation, and social systems while remaining faithful to structural controllability.

In summary, research on duplex (and broader multilayer) controllability has progressed from foundational modeling and single-layer structural controllability to two-layer mechanisms shaped by coupling rules, through alternative multilayer surrogates (e.g., dominating-set and conserved-path perspectives), and finally to ensemble-, energy-, and application-driven trends across neuroscience, biology, CPS/power, transportation, and social systems. As consolidated in Table 1, none of these lines simultaneously (i) preserve *per-layer* minimal driver size by remaining within the space of *maximum matchings* and (ii) *minimize the union* of driver nodes across layers, nor do they provide constructive optimality certificates for that union objective. This gap motivates our matching-theoretic formulation and cross-layer alternating-path mechanism that coordinates layer-wise maximum matchings to reduce the union while certifying optimality when no cross-layer augmenting structure exists.

3 Preliminaries and Problem Definition

In this section, we establish the theoretical foundation for our work. We begin by introducing basic graph notations and the principles of structural controllability [63], [64] within a linear time-invariant framework. We then review the core graph-theoretic concepts of matchings and alternating paths [65], [66], [67] that allow us to formally define the problem of budget-preserving union contraction for driver sets in duplex networks.

TABLE 1
SUMMARY OF RELATED WORKS: focus, cross-layer assumption, and relation to our goal (per-layer minimality via maximum matching; minimize union of drivers across layers).

Work (examples)	Problem focus	Cross-layer actuation	Relation to our objective
Foundations of multilayer networks [24], [25]	Modeling/metrics	Not applicable	Establish multilayer representation; no control objective.
Liu *et al.* [5]	Single-layer structural controllability (matching → MDS)	None	Baseline: unmatched vertices are drivers; no cross-layer coordination.
Menichetti *et al.* [31]	Duplex controllability via matching	*Identical* drivers across layers	Enforces maximum reuse but overrestrictive; no union optimization under per-layer minimality.
Pósfai *et al.* [32]	Duplex with multi-timescale; inputs on one layer	One-layer actuation	Clarifies timing effects; no union-of-drivers optimization.
Zhang *et al.* [37]	Duplex interconnection/driver placement strategy	One-layer or asymmetric	Peripheral connectors/drivers can help; still no union objective.
Nacher *et al.* [48]	Multilayer control via dominating sets	Not matching-based	Scalable multilayer control surrogate; no per-layer matching minimality.
Wang *et al.* [49]	Conserved control paths across layers	Path-level commonality	Seeks common structures; does not minimize driver union or ensure per-layer minimality.
Zheng *et al.* [55]	Minimum-union interventions in multilayer biology (nonlinear)	Union minimized	Closest spirit; outside matching-based structural controllability.
Single-layer alternating-path toolbox [11], [18], [19], [50], [51]	Enumerate/reconfigure MDSs while preserving minimality	Not applicable	Provides primitives to *align* layer-wise MDSs via alternating exchanges.
Applications: brain, CPS/power, transportation [4], [54], [56], [57], [58], [59]	Domain evidence for actuator economy	Prefer reuse across layers/modalities	Motivate our union-minimization under per-layer minimality.

3.1 Foundational Concepts

We first define the basic notation used throughout this paper. A summary is provided in Table 2. The terms "node" and "vertex", as well as "edge" and "link", are used interchangeably.

We adopt the multilayer terminology of Kivelä et al. and De Domenico et al.: a duplex is a two-layer multiplex sharing a (possibly aligned) node set; revisehere we focus on the aligned-node setting with only intra-layer edges (i.e., no explicit inter-layer state-transition links), where layers are coupled solely through shared actuator placement on the common node set [24], [68].

The dynamics of many networked systems can be modeled using a linear time-invariant (LTI) framework. For a network of N nodes, the state of the system is described by a vector $\mathbf{x}(t) \in \mathbb{R}^N$, and its evolution is governed by:

$$\frac{d\mathbf{x}(t)}{dt} = A\mathbf{x}(t) + B\mathbf{u}(t) \tag{1}$$

where $A \in \mathbb{R}^{N\times N}$ is the system's interaction matrix, and $\mathbf{u}(t) \in \mathbb{R}^M$ is the vector of M control inputs applied to a specific set of *driver nodes* via the input matrix $B \in \mathbb{R}^{N\times M}$.

3.1.1 Duplex LTI Dynamics and Simultaneous Layer Control

We next specialize the LTI model in (1) to duplex networks. Consider two directed layers $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ defined on the same node set V, where $|V| = N$. Let $\mathbf{x}^{(\ell)}(t) \in \mathbb{R}^N$ denote the state vector of layer $\ell \in \{1, 2\}$, and define the stacked duplex state $\mathbf{x}(t) = [\mathbf{x}^{(1)}(t)^T, \mathbf{x}^{(2)}(t)^T]^T \in \mathbb{R}^{2N}$. In the regime where the two layers are dynamically isolated (i.e., there are no explicit inter-layer state-transition links), the duplex dynamics admit the block-diagonal form:

$$\frac{d\mathbf{x}(t)}{dt} = \begin{pmatrix} A^{(1)} & \mathbf{0} \\ \mathbf{0} & A^{(2)} \end{pmatrix} \mathbf{x}(t) + \begin{pmatrix} B^{(1)} & \mathbf{0} \\ \mathbf{0} & B^{(2)} \end{pmatrix} \mathbf{u}(t), \tag{2}$$

where $A^{(\ell)} \in \mathbb{R}^{N\times N}$ respects the sparsity pattern of layer G_ℓ. The input vector $\mathbf{u}(t) = [\mathbf{u}^{(1)}(t)^T, \mathbf{u}^{(2)}(t)^T]^T$ collects layer-wise inputs, where $\mathbf{u}^{(\ell)}(t) \in \mathbb{R}^{k_\ell}$ represents the k_ℓ control signals applied to layer ℓ, and $B^{(\ell)} \in \mathbb{R}^{N\times k_\ell}$ maps these inputs to the driver nodes. For block-diagonal A, controllability of the duplex reduces to controllability of each layer-wise pair $(A^{(1)}, B^{(1)})$ and $(A^{(2)}, B^{(2)})$. Accordingly, in this paper, "simultaneous control" implies selecting input locations on the shared node set such that *each layer* is structurally controllable. We measure the deployment cost by the number of *unique* actuated nodes, i.e., the union of the layer-wise driver sets.

In many complex systems, the exact weights of interactions are unknown. This motivates the use of *structural controllability* [6], which depends only on the network's topology.

Remark 1 (Assumption on Generic Weights). The structural controllability framework relies on the *generic rank assumption* [6], [64]: the non-zero entries of $A^{(\ell)}$ are assumed to be algebraically independent free parameters. Under this assumption, a system is controllable for almost all weight combinations if it satisfies the topological conditions derived from graph matching. While real-world weights may exhibit correlations, this framework provides the standard topological lower bound for driver requirements.

TABLE 2
Nomenclature and Symbols

Symbol	Description
A. General Network & Dynamics	
$G = (V, E)$	Directed graph with node set V and edge set E
N, M	Number of nodes ($\|V\|$) and inputs
ℓ	Layer index, $\ell \in \{1, 2\}$
$A^{(\ell)}, B^{(\ell)}$	State and Input matrices for layer ℓ
$\mathbf{x}(t), \mathbf{u}(t)$	State vector and control input vector
k_ℓ	Fixed driver budget for layer ℓ
B. Structural Controllability & Matching	
$\mathcal{B}$	Bipartite representation defined by G
$E_\mathcal{B}$	Edge set of the bipartite graph $\mathcal{B}$
V^+, V^-	Sets of out-nodes (source) and in-nodes (target)
v^+, v^-	Copies of node v in V^+ and V^-
$\mathcal{M}, \mathcal{M}^\star$	Matching and Maximum Matching in $\mathcal{B}$
$D(\mathcal{M})$	Driver node set determined by $\mathcal{M}$
$\mathbb{M}_\ell(k_\ell)$	Feasible set of max matchings for layer ℓ
$p_\mathcal{M}$	$\mathcal{M}$-alternating path
$\triangle$	Symmetric difference operator
C. Union Minimization Problem	
Ω	Feasible configuration space $\mathbb{M}_1 \times \mathbb{M}_2$
$U(\mathcal{M}_1, \mathcal{M}_2)$	Union Driver Set (UDS), $D_1 \cup D_2$
$U^\star$	Minimum UDS over feasible matchings
CDS	Consistently Driven Set, $D_1 \cap D_2$
CMS	Consistently Matched Set, $(V \setminus D_1) \cap (V \setminus D_2)$
DD_ℓ	Difference-Driver Set (drivers unique to layer ℓ)
$\Delta(\mathcal{M}_1, \mathcal{M}_2)$	Difference Mass, $\|\mathrm{DD}_1\| + \|\mathrm{DD}_2\|$
D. CLAP Framework	
$\mathcal{P}$	Cross-Layer Augmenting Path (CLAP)
$u \xrightarrow{\ell} v$	Admissible segment in layer ℓ
$W_\ell(\mathcal{P})$	Set of witness path edges in layer ℓ
$\delta(v)$	State difference vector of node v
$\mathcal{K}$	Layer-labeled meta-graph for optimality proof
$h, \bar{h}$	Length and average length of a CLAP
E. Metrics & Experiments	
ΔN_D	Absolute reduction in UDS size
$\Delta N_D^{\mathrm{opt}}$	Optimization gain (CLAP-S vs. RSU)
R_{opt}	Relative optimization rate (%)
$\langle k \rangle, \rho$	Average degree and edge overlap ratio
S	Sampling budget for randomized baseline
W, E	Controllability Gramian and control energy metric
$p, J(p)$	Edge removal fraction and Jaccard similarity

The structural controllability of a network $G = (V, E)$ is determined by the matching properties of its bipartite representation [5].

Definition 1 (Bipartite Representation)**.** For a directed graph $G = (V, E)$, its **bipartite representation** is a bipartite graph $\mathcal{B} = (V^+ \cup V^-, E_\mathcal{B})$. The two disjoint vertex sets are copies of V: $V^+ = \{v^+ \mid v \in V\}$ and $V^- = \{v^- \mid v \in V\}$. For every directed edge $(u, v) \in E$, there is a corresponding edge $(u^+, v^-) \in E_\mathcal{B}$. This construction is the standard reduction used in structural controllability [5]. Fig.1B illustrates the concept of Bipartite Representation.

A *matching* $\mathcal{M}$ in $\mathcal{B}$ is a subset of $E_\mathcal{B}$ where no two edges share a common vertex. The set of vertices incident to an edge in $\mathcal{M}$ is the saturated vertex set $V_\mathcal{M}$.

Theorem 1 (Minimum Input Theorem (*cf.* [5]))**.** *The minimum number of driver nodes N_D required to structurally control a network G is determined by the cardinality of a maximum matching $\mathcal{M}^\star$ in $\mathcal{B}$:*

$$N_D = \max(1, N - |\mathcal{M}^\star|). \tag{3}$$

*Assuming $N - |\mathcal{M}^\star| \geq 1$, we define the **driver budget** k_ℓ for layer ℓ as the invariant quantity $k_\ell = N - |\mathcal{M}_\ell^\star|$.*

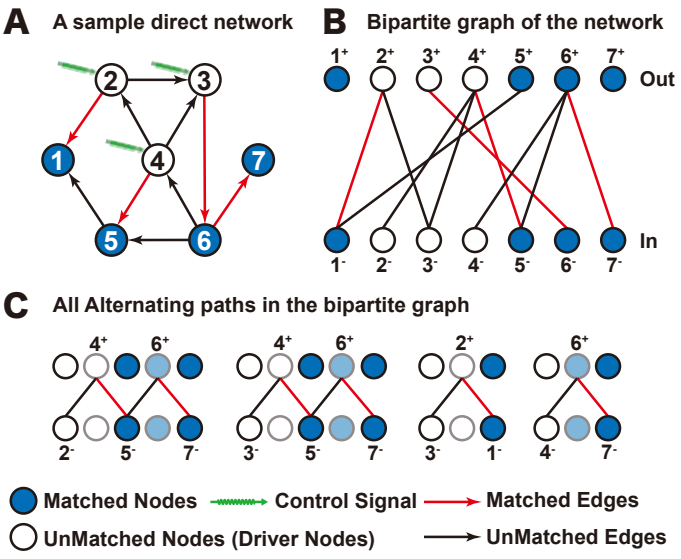

Fig. 1. Illustration of foundational concepts for structural controllability. (A) A sample directed network $G = (V, E)$. (B) The bipartite representation $\mathcal{B}$ of the network. A maximum matching $\mathcal{M}^\star$ is shown with red edges. The vertices in V^- that are unmatched by $\mathcal{M}^\star$ (white circles) correspond to the nodes in the Minimum Driver Set (MDS), which is $D(\mathcal{M}^\star) = \{2, 3, 4\}$. (C) Examples of $\mathcal{M}^\star$-alternating paths. Such paths allow for the transformation between different maximum matchings, yielding different MDS configurations of the same minimum size.

Remark 2 (Minimal Budgets as the Binding Constraint). We restrict our theoretical development to the *minimal* driver budgets k_ℓ derived from maximum matchings. This represents the most constrained and structurally challenging regime. Since structural controllability is monotonic (adding drivers preserves controllability [5]), any solution for the minimal budget k_ℓ can be trivially extended to a relaxed budget $\tilde{k}_\ell > k_\ell$ by appending arbitrary nodes. Therefore, the minimal-budget problem constitutes the combinatorial kernel of the general budget-constrained problem.

Definition 2 (Driver Set)**.** For a matching $\mathcal{M}$, the corresponding **driver set** $D(\mathcal{M})$ consists of nodes whose target vertices (V^-) are unmatched:

$$D(\mathcal{M}) = \{\, v \in V : \; v^- \notin V_\mathcal{M} \,\}. \tag{4}$$

Any driver set derived from a maximum matching $\mathcal{M}^\star$ is a *Minimum Driver Set (MDS)*. In the context of duplex networks, we explicitly denote the driver set of layer ℓ as $D_\ell(\mathcal{M}_\ell) \equiv D(\mathcal{M}_\ell)$, or simply D_ℓ, where $\mathcal{M}_\ell$ is a maximum matching for layer ℓ.

Throughout this work, we treat the budgets k_1 and k_2 as immutable structural invariants. The optimization task is not to change k_ℓ, but to adjust the specific composition of the driver sets D_ℓ via matching reconfiguration.

3.2 The Driver Exchange Principle

While the driver budget k_ℓ is invariant, the driver set itself is structurally plastic. To exploit this, we require an atomic mechanism to reconfigure drivers within a layer without violating the budget. This mechanism relies on alternating paths in the bipartite graph $\mathcal{B}$, treated here as undirected.

Definition 3 (Alternating Path)**.** A path $\mathcal{P}$ in $\mathcal{B}$ is **$\mathcal{M}$-alternating** if its edges alternate between $E \cap \mathcal{M}$ and $E \setminus \mathcal{M}$. Fig.1C shows all the alternating paths currently present in the sample network, totaling 4.

An $\mathcal{M}$-alternating path can be used to modify a matching $\mathcal{M}$ into a new matching $\mathcal{M}'$ of the same size. [69] This operation, which forms the basis of our approach, allows for the reconfiguration of the driver set while preserving its cardinality. [11], [50], [51], [70]

Theorem 2 (Driver Exchange Principle (*cf.* [11], [50], [51]))**.** *Let $\mathcal{M}$ be a matching. Let $s \in D(\mathcal{M})$ be a driver node and $t \in V \setminus D(\mathcal{M})$ be a non-driver node. There exists an $\mathcal{M}$-alternating path p from s^- to t^- in $\mathcal{B}$ if and only if the matching $\mathcal{M}' = \mathcal{M} \triangle E(p)$, where $E(p)$ is the set of edges in p, satisfies*

(i) $|\mathcal{M}'| = |\mathcal{M}|$, *which implies* $|D(\mathcal{M}')| = |D(\mathcal{M})|$, *and*
(ii) $D(\mathcal{M}') = (D(\mathcal{M}) \setminus \{s\}) \cup \{t\}$.

Proof Sketch: Since s^- is unmatched and t^- is matched, the path $\mathcal{P}$ must begin with a non-matching edge and end with a matching edge to satisfy the alternation constraint. This implies $\mathcal{P}$ has even length. The symmetric difference operation toggles the membership of edges along $\mathcal{P}$.

(i) **Internal vertices:** Incident to exactly one edge in $\mathcal{M}$ and one in $E_\mathcal{B} \setminus \mathcal{M}$. After toggling, they remain incident to exactly one matching edge; hence, their matched status is preserved.
(ii) **Endpoints:** s^- gains a matching edge (becoming non-driver), while t^- loses its matching edge (becoming driver).

Since the total number of matched edges is conserved, the operation is budget-preserving. Formal proofs on parity and validity are detailed in **Supplementary Section 1**. □

Theorem 2 provides a rigorous *primitive* operation: it allows us to swap a driver node s with a non-driver node t while strictly preserving the driver budget. In the next section, we utilize this freedom to coordinate driver sets across two layers.

3.3 Problem Formulation: Budget-Preserving Union Contraction

Ideally, a duplex network would be controlled by a minimal set of shared actuators. However, treating layers independently often yields disjoint driver sets. Leveraging the reconfiguration freedom established in Section 3.2, we now formulate the problem of coordinating these sets.

Definition 4 (Feasible Configuration Space)**.** Let $\mathbb{M}_\ell(k_\ell)$ denote the set of all valid maximum matchings for layer ℓ satisfying the minimum budget k_ℓ. The **feasible configuration space** Ω for the duplex network is defined as the Cartesian product:

$$\Omega := \mathbb{M}_1(k_1) \times \mathbb{M}_2(k_2). \tag{5}$$

An element $(\mathcal{M}_1, \mathcal{M}_2) \in \Omega$ represents a valid dual-layer control configuration where each layer maintains structural controllability.

For any configuration in this space, the deployment cost is determined by the overlap of the layer-specific driver sets.

Definition 5 (Union Driver Set)**.** For a state $(\mathcal{M}_1, \mathcal{M}_2) \in \Omega$, the **Union Driver Set (UDS)** is defined as the set of unique nodes required to actuate both layers:

$$U(\mathcal{M}_1, \mathcal{M}_2) = D_1(\mathcal{M}_1) \cup D_2(\mathcal{M}_2). \tag{6}$$

As illustrated in Fig.2, uncoordinated selection of matchings leads to redundancy. Our goal is to navigate Ω to identify the configuration with maximum overlap.

Problem 1 (Budget-Preserving UDS Contraction)**.** Given a duplex network with minimum budgets (k_1, k_2), find a configuration $(\mathcal{M}_1^\star, \mathcal{M}_2^\star) \in \Omega$ that minimizes the cardinality of the union driver set:

$$(\mathcal{M}_1^\star, \mathcal{M}_2^\star) = \underset{(\mathcal{M}_1, \mathcal{M}_2) \in \Omega}{\arg\min} \big| D_1(\mathcal{M}_1) \cup D_2(\mathcal{M}_2) \big|. \tag{7}$$

We denote the minimum union cardinality as $|U^\star|$.

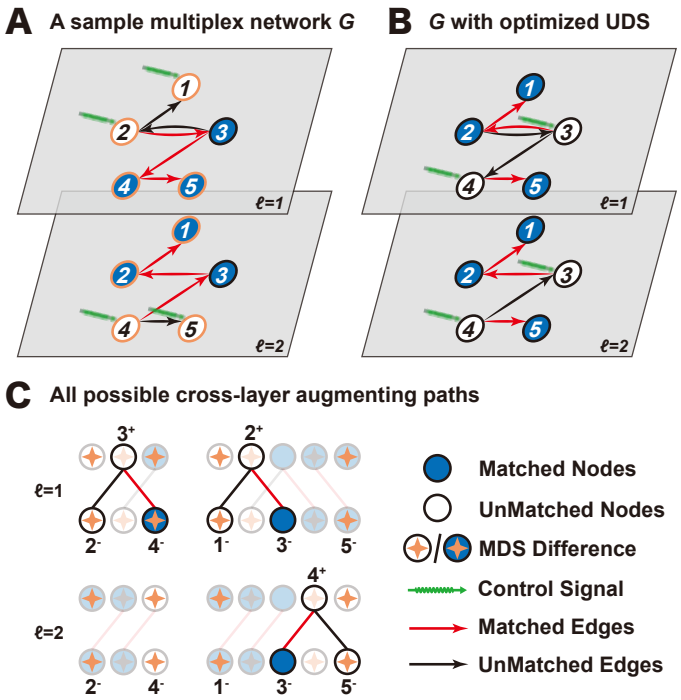

Fig. 2. Illustration of the budget-preserving UDS contraction problem. (A) An initial state of a duplex network with driver budgets $k_1 = 2, k_2 = 2$. The chosen $\mathcal{M}_\ell$ yield driver sets $D_1 = \{1, 2\}$ and $D_2 = \{4, 5\}$. The initial UDS is $\{1, 2, 4, 5\}$, with size 4. (B) A contracted state for the same network. By reconfiguring the $\mathcal{M}_\ell$ within the same budgets, the new driver sets are $D_1' = \{3, 4\}$ and $D_2' = \{3, 4\}$. The resulting UDS is $\{3, 4\}$, with size 2. This represents a more efficient alignment of driver nodes.

The primary challenge is that Ω grows combinatorially with network size, rendering brute-force search infeasible.

Remark 3 (Generalization to Arbitrary Budgets). Our formulation focuses on the *minimum* budgets k_ℓ (the "tightest" constraint regime). This is not a limitation but a focus on the combinatorial kernel of the problem. Due to the monotonicity of structural controllability [5], any superset of a driver set remains valid. Therefore, for any relaxed engineering budgets $K_\ell \geq k_\ell$, the optimal union size is simply $\max(|U^\star|, K_1, K_2)$. Constructively, the global optimum for relaxed budgets is obtained by first solving Problem 1 to find the minimal core $U^\star$, and then augmenting it with arbitrary shared nodes until the budgets K_ℓ are satisfied. Thus, the algorithm presented here constitutes the fundamental solver for any budget specification.

3.4 Objective Transformation: Minimizing Difference Mass

To systematically reduce the size of the UDS we partition the node set V based on a given state $(\mathcal{M}_1, \mathcal{M}_2)$. Let $D_1 = D_1(\mathcal{M}_1)$ and $D_2 = D_2(\mathcal{M}_2)$.

Definition 6 (Node Partition)**.** The node set V is partitioned into four mutually disjoint subsets:

(i) The **Consistently Driven Set** (CDS): $\text{CDS} = D_1 \cap D_2$.
(ii) The **Consistently Matched Set** (CMS): $\text{CMS} = (V \setminus D_1) \cap (V \setminus D_2)$.
(iii) The **Difference-Driver Set 1** (DD_1): $\text{DD}_1 = D_1 \setminus D_2$.
(iv) The **Difference-Driver Set 2** (DD_2): $\text{DD}_2 = D_2 \setminus D_1$.

Nodes in DD_1 and DD_2 act as drivers in only one layer, representing control redundancy.

To quantify this inefficiency, we measure the size of the symmetric difference between the driver sets.

Definition 7 (Difference Mass)**.** The **difference mass** of a state $(\mathcal{M}_1, \mathcal{M}_2)$ is defined as the total number of unaligned driver nodes:

$$\Delta(\mathcal{M}_1, \mathcal{M}_2) = |\text{DD}_1| + |\text{DD}_2|. \tag{8}$$

Crucially, under the fixed-budget constraint, the size of the UDS is strictly governed by the difference mass.

Proposition 1 (Equivalence of Objectives)**.** *For any state $(\mathcal{M}_1, \mathcal{M}_2) \in \Omega$, the size of the UDS is given by*

$$|U(\mathcal{M}_1, \mathcal{M}_2)| = \frac{k_1 + k_2 + \Delta(\mathcal{M}_1, \mathcal{M}_2)}{2}. \tag{9}$$

Proof: By the principle of inclusion-exclusion, the sum of the layer budgets is:

$$k_1 + k_2 = |D_1| + |D_2| = 2|\text{CDS}| + |\text{DD}_1| + |\text{DD}_2| = 2|\text{CDS}| + \Delta. \tag{10}$$

The union size is the sum of the disjoint components:

$$|U| = |\text{CDS}| + |\text{DD}_1| + |\text{DD}_2| = |\text{CDS}| + \Delta. \tag{11}$$

Expressing $|\text{CDS}| = |U| - \Delta$ from (11) and substituting it into (10) yields:

$$k_1 + k_2 = 2(|U| - \Delta) + \Delta = 2|U| - \Delta.$$

Rearranging this equation gives $|U| = (k_1 + k_2 + \Delta)/2$. Since k_1 and k_2 are structural invariants, minimizing $|U|$ is strictly equivalent to minimizing Δ. □

Proposition 1 is pivotal. It bridges the gap between engineering goals and graph theory: the optimization of actuator reuse (minimizing $|U|$) is transformed into a search for graph reconfigurations that eliminate nodes from the difference sets DD_1 and DD_2. This algebraic transformation sets the stage for our Cross-Layer Augmenting Path (CLAP) framework.

4 The Cross-Layer Augmenting Path Framework and Algorithm

As established in Section 3.4, minimizing the union size is equivalent to minimizing the difference mass Δ. We now introduce the Cross-Layer Augmenting Path (CLAP) framework, which constructs directed paths in the configuration space to systematically eliminate nodes from the difference sets DD_1 and DD_2.

4.1 Cross-Layer Admissible Segments

To systematically reduce Δ, we seek a directed path $\mathcal{P} = (v_0, v_1, \ldots, v_k)$ in the configuration space. We define the roles of the nodes in such a path as follows:

(i) **Source (v_0):** A node starting in DD_1 that initiates the reconfiguration.
(ii) **Target (v_k):** A node starting in DD_2 that terminates the path.
(iii) **Relays ($v_i, 0 < i < k$):** Intermediate nodes that facilitate the transfer of driver status between layers.

The connections between these nodes are governed by the specific driver exchange rules defined below.

Definition 8 (Admissible Segment)**.** Given a state $(\mathcal{M}_1, \mathcal{M}_2)$, an ordered pair (u, v) is an **admissible segment in layer** ℓ, denoted $u \xrightarrow{\ell} v$, if it supports a directional driver displacement:

(i) **Layer 1 (Forward Push):** Requires $u \in D_1, v \notin D_1$. Moves a driver from u to v.
(ii) **Layer 2 (Backward Pull):** Requires $u \notin D_2, v \in D_2$. Moves a driver from v to u.

The existence of such segments is guaranteed by the Alternating Path Principle (Theorem 2).

Remark 4 (Symmetry Convention). We fix the search direction from DD_1 to DD_2 (Layer 1 Push / Layer 2 Pull) to break symmetry without loss of generality. A reduction path from DD_2 to DD_1 is mathematically equivalent to the reverse traversal of the defined path.

For a path to be valid, consecutive segments must alternate layers ($u \xrightarrow{1} v \xrightarrow{2} w$). This alternation imposes strict feasibility conditions on the relay nodes.

Lemma 1 (Relay Feasibility Condition)**.** *A node v can serve as a relay if and only if it belongs to a specific consistent set, determined by the sequence of layers:*

(i) ***L1-to-L2 Relay*** ($u \xrightarrow{1} v \xrightarrow{2} w$)***:*** *Must be in CMS.*
(ii) ***L2-to-L1 Relay*** ($u \xrightarrow{2} v \xrightarrow{1} w$)***:*** *Must be in CDS.*

Proof: For case (i), the incoming segment $u \xrightarrow{1} v$ requires $v \notin D_1$ (destination of a push). The outgoing segment $v \xrightarrow{2} w$ requires $v \notin D_2$ (destination of a pull). Thus, $v \in (V \setminus D_1) \cap (V \setminus D_2) =$ CMS. Case (ii) follows symmetrically: v must be a source for L1 ($v \in D_1$) and a source for L2 ($v \in D_2$), implying $v \in$ CDS. This constraint ensures that relay nodes act as *lossless converters*. A **CMS relay** starts with no drivers; applying the path adds a driver in L1 (from u) and adds a driver in L2 (from w), transitioning it to CDS. Conversely, a **CDS relay** starts with drivers in both layers; the path removes both, transitioning it to CMS. In both cases, the relay node remains "consistent" (either both or neither), ensuring that the net reduction of difference mass Δ occurs exclusively at the Source and Target endpoints. □

4.2 The CLAP Mechanism and Gain

By chaining admissible segments, we construct the Cross-Layer Augmenting Path, a directed structure designed to transport driver surplus from DD_1 to a deficiency in DD_2.

Definition 9 (Cross-Layer Augmenting Path)**.** A **Cross-Layer Augmenting Path** is a sequence of admissible segments

$$\mathcal{P} = (v_0 \xrightarrow{\ell_1} v_1 \xrightarrow{\ell_2} v_2 \cdots \xrightarrow{\ell_k} v_k)$$

satisfying three structural properties:

(i) **Boundary Conditions:** The path originates in DD_1 ($v_0 \in DD_1$) and terminates in DD_2 ($v_k \in DD_2$).
(ii) **Layer Alternation:** Consecutive segments act on different layers: $\ell_{i+1} \neq \ell_i$ for all $1 \leq i < k$.
(iii) **Simplicity:** All nodes $v_0, \ldots, v_k$ are distinct.

The execution of a CLAP involves applying the symmetric difference updates along the witness paths of all constituent segments. This operation yields a strictly improved configuration.

Theorem 3 (CLAP Gain Theorem)**.** *Let $\mathcal{P}$ be a valid CLAP for the state $(\mathcal{M}_1, \mathcal{M}_2)$. Applying the updates $\mathcal{M}_\ell \leftarrow \mathcal{M}_\ell \triangle W_\ell(\mathcal{P})$ transforms the system into a new state $(\mathcal{M}'_1, \mathcal{M}'_2)$ such that:*

(i) ***Budget Invariance:*** $|D'_\ell| = |D_\ell| = k_\ell$.
(ii) ***Mass Reduction:*** $\Delta(\mathcal{M}'_1, \mathcal{M}'_2) = \Delta(\mathcal{M}_1, \mathcal{M}_2) - 2$.
(iii) ***Union Contraction:*** $|U'| = |U| - 1$.

Proof: **(i) Budget:** By Definition 8, each segment is witnessed by an alternating path. By Theorem 2, symmetric difference along such paths preserves the cardinality of matchings, keeping k_ℓ invariant.

(ii) Mass Reduction: We track the transition of node roles: **Source ($v_0 \in DD_1$):** The first segment either removes an L1 driver (Push) or adds an L2 driver (Pull). In either case, v_0 transitions to a consistent set (CMS or CDS), reducing Δ by 1. **Target ($v_k \in DD_2$):** Symmetrically, the final segment either adds an L1 driver or removes an L2 driver. v_k transitions to a consistent set (CDS or CMS), reducing Δ by 1. **Relays (v_i):** By the Relay Feasibility Condition (Lemma 1), intermediate nodes act as lossless converters (transitioning CMS $\leftrightarrow$ CDS). Their contribution to Δ remains 0. The net change is exactly $-1 - 1 + 0 = -2$.

(iii) Union Contraction: From Proposition 1, $|U| = (k_1 + k_2 + \Delta)/2$. A decrease of 2 in Δ directly implies a decrease of 1 in $|U|$. □

Theorem 3 establishes the CLAP as a deterministic descent operator. The optimization process is thus reduced to iteratively finding and applying these paths until the difference mass can no longer be reduced.

4.3 Theoretical Guarantees and Iterative Strategy

The CLAP mechanism provides a local improvement operator. A fundamental question remains: Does the iterative application of local improvements guaranty convergence to a *global* optimum, or can the algorithm get trapped in a local minimum?

We answer this by establishing the equivalence between local stability and global optimality.

Definition 10 (CLAP-Stability)**.** A configuration $(\mathcal{M}_1, \mathcal{M}_2) \in \Omega$ is **CLAP-stable** if it admits no feasible Cross-Layer Augmenting Paths.

Theorem 4 (CLAP-or-Optimal)**.** *A configuration $(\mathcal{M}_1, \mathcal{M}_2)$ minimizes the UDS size over the feasible space Ω if and only if it is CLAP-stable.*

Proof Sketch: The proof generalizes Berge's Lemma [60] to the duplex setting. Consider a stable state S and a hypothetical state S^* with strictly lower difference mass ($\Delta^* < \Delta$). The symmetric difference between the matchings in S and S^* induces a graph structure composed of disjoint paths and cycles. Internal nodes in these components preserve mass (Lemma 1). For the total mass to decrease, there must exist at least one path component connecting a surplus node in DD_1 to a deficiency node in DD_2. Crucially, such a path component is topologically isomorphic to a CLAP. Thus, if S is not optimal, it must contain a CLAP. Conversely, if S admits no CLAP (stable), no such reducing structure exists, implying S is globally optimal. Formal construction of the difference meta-graph is detailed in **Supplementary Section 3**. □

This theorem identifies CLAPs as a **complete improvement primitive**, justifying a greedy iterative strategy. To implement this strategy efficiently, we enforce a shortest-path policy.

Proposition 2 (Shortest-Path Validity)**.** *By prioritizing the **shortest** CLAP (in terms of the number of segments) at each iteration, we ensure two properties:*

(i) ***Computational Efficiency:*** *The path can be found via Breadth-First Search (BFS), avoiding exponential enumeration.*
(ii) ***Update Atomicity:*** *As proved in **Supplementary Section 2**, the witness paths constituting a shortest CLAP are guaranteed to be edge-disjoint within each layer. This allows the multi-segment reconfiguration to be executed as a single atomic transaction without internal edge conflicts.*

4.4 The CLAP-S Algorithm

Guided by the optimality certificate and the shortest-path validity, we present CLAP-S (**C**ross-**L**ayer **A**ugmenting **P**ath **S**earch). The algorithm (Algorithm 4.4.1) iteratively reduces the difference mass until the system reaches a CLAP-stable state.

4.4.1 Algorithm Description

The procedure consists of three phases:

Initialization: Compute initial maximum matchings $\mathcal{M}_1, \mathcal{M}_2$ using the Hopcroft-Karp algorithm. Construct the initial partition sets ($\mathrm{DD}_1, \mathrm{DD}_2, \mathrm{CMS}, \mathrm{CDS}$).

Iterative Improvement: Execute a loop that performs a layer-alternating BFS (Algorithm 4.4.1) to find the shortest CLAP $\mathcal{P}$ from DD_1 to DD_2. If found, apply the atomic update $\mathcal{M}_\ell \leftarrow \mathcal{M}_\ell \triangle W_\ell(\mathcal{P})$ and update the partition sets. This reduces Δ by exactly 2.

Termination: If the BFS exhausts the search space without finding a target in DD_2, the current state is stable. By Theorem 4, the algorithm terminates with the global minimum.

Algorithm 1 — CLAP-S: Iterative Optimization

Require: Duplex layers G_1, G_2.
Compute initial max matchings $\mathcal{M}_1, \mathcal{M}_2$.
Identify sets $\mathrm{DD}_1, \mathrm{DD}_2, \mathrm{CMS}, \mathrm{CDS}$.
while true do
  $\mathcal{P} \leftarrow$ FINDSHORTESTCLAP$(\mathcal{M}_1, \mathcal{M}_2)$
  if $\mathcal{P}$ is null **then**
    break ▷ *Optimality Certificate (Thm 4)*
  end if
  $(\mathcal{M}_1, \mathcal{M}_2) \leftarrow$ APPLYATOMICUPDATE$(\mathcal{P}, \mathcal{M}_1, \mathcal{M}_2)$
  Update sets $\mathrm{DD}_1, \mathrm{DD}_2, \mathrm{CMS}, \mathrm{CDS}$
end while
return $(\mathcal{M}_1, \mathcal{M}_2)$ ▷ *Optimal State*

The FINDSHORTESTCLAP subroutine (Algorithm 4.4.1) implements the layer-alternating BFS. It implicitly explores the configuration space by generating admissible segments on-the-fly.

Algorithm 2 — FINDSHORTESTCLAP: Layer-Alternating BFS

Require: Current state $\mathcal{M}_1, \mathcal{M}_2$, Sets $\mathrm{DD}_1, \mathrm{DD}_2, \mathrm{CMS}, \mathrm{CDS}$.
$Q \leftarrow$ Queue containing $(\mathrm{DD}_1, 1)$ and $(\mathrm{DD}_1, 2)$ ▷ *State (S, ℓ): reached all nodes in frontier S, next segment leaves in layer ℓ*
Mark $(s, 1), (s, 2)$ for all $s \in \mathrm{DD}_1$ as visited
initialize pred$[\cdot,\cdot]$, predLayer$[\cdot,\cdot]$
while Q is not empty **do**
  $(S, \ell) \leftarrow Q$.pop() ▷ *FIFO ensures shortest segments*
  $(R, \omega) \leftarrow$ ALTREACH$(S \mid \mathcal{M}_\ell)$ ▷ *(Here $\omega(v) \in S$ witnesses a segment source for each $v \in R$.)*
  Check Targets: $T \leftarrow R \cap \mathrm{DD}_2$
  if $T \neq \varnothing$ **then**
    pick any $v \in T$; $u \leftarrow \omega(v)$
    return UNROLLSEGMENTS$((u, \ell), v, \mathrm{pred}, \mathrm{predLayer})$
  end if
  Check Relays:
  $R_{\text{relay}} \leftarrow (R \cap \mathrm{CMS})$ if $\ell = 1$ else $(R \cap \mathrm{CDS})$
  for all $w \in R_{\text{relay}}$ unvisited **do**
    pred$[w, 3-\ell] \leftarrow \omega(w)$; predLayer$[w, 3-\ell] \leftarrow \ell$ ▷ *mark visited*
    Enqueue $(\{w\}, 3-\ell)$ ▷ *Switch layer for next segment*
  end for
end while
return null

Remark 5 (Auxiliary routines). For brevity, Algorithms 1 and 2 treat several standard subroutines as black boxes. (i) ALTREACH performs a multi-source alternating BFS in layer ℓ and returns the reachable set, together with an origin map ω that witnesses which source in the current frontier generated each reachability. (ii) UNROLLSEGMENTS backtracks the predecessor pointers $(\mathrm{pred}, \mathrm{predLayer})$ to recover the *segment endpoints* of the shortest CLAP. (iii) APPLYATOMICUPDATE executes the CLAP by realizing each segment via a shortest alternating path (ALTPATH) under the *current* matching and updating the matching via the symmetric difference. Full pseudocode and implementation details of these auxiliary routines are provided in **Supplementary Sec. 4**.

4.4.2 Complexity Analysis

We derive the time complexity by decomposing the procedure into initialization, iteration bounds, and search costs. To facilitate physical interpretation, we introduce the network size N, total edges $E_{\text{tot}} = E_1 + E_2$, and average degree $\langle k \rangle \approx E_{\text{tot}}/N$.

The complexity comprises three components: **Initialization:** Calculating maximum matchings via Hopcroft-Karp requires $O(E_{\text{tot}}\sqrt{N})$. **Iteration Count ($I$):** The loop is governed by the difference mass Δ. Each successful CLAP reduces Δ by 2. In the worst case (disjoint layers), $\Delta_0 \approx N$ implies that the number of iterations is bounded by $I_{\max} = N/2$. **Search Cost per Iteration (C):** The FINDSHORTESTCLAP routine performs a multi-source BFS. In the worst case, the search frontier expands until it traverses the entire graph before finding a target or determining infeasibility. Thus, the per-iteration cost is bounded by the total edges: $C_{\text{worst}} = O(E_{\text{tot}}) = O(N\langle k \rangle)$.

Aggregating these, the theoretical worst-case complexity is as follows:

$$T_{\text{worst}} \approx \underbrace{I_{\max}}_{O(N)} \times \underbrace{C_{\text{worst}}}_{O(E_{\text{tot}})} = O(N \cdot E) = O(N^2\langle k \rangle). \tag{12}$$

This quadratic scaling with N suggests that for massive networks ($N \sim 10^6$), the algorithm might be computationally expensive.

However, the worst-case assumption—that BFS traverses the entire graph—rarely holds. The actual search cost depends on the **CLAP length** h (number of segments). For a BFS of depth h starting from a set of difference nodes DD, the number of visited edges E_{visit} is not the total E_{tot}, but is bounded by the local expansion:

$$C_{\text{eff}}(h) \approx O\big(\min(E_{\text{tot}},\ |\mathrm{DD}| \cdot \langle k \rangle^h)\big). \tag{13}$$

Here, $|\mathrm{DD}| \cdot \langle k \rangle^h$ represents the volume of the search sphere expanding from the difference set.

We can now contrast the theoretical and effective complexities through the lens of path length h: **Theoretical Regime ($h \to \infty$):** When conflicts are hard to resolve, the search penetrates deep into the graph. The effective cost saturates to the graph size ($C_{\text{eff}} \to E_{\text{tot}}$), recovering the worst-case bound $T \to O(N \cdot E_{\text{tot}})$. **Practical Regime ($h \approx 1$):** As empirically observed in **Supplementary Fig. 5**, the average path length is extremely short ($\bar{h} \approx 1.2$). This implies that the "conflict resolution" is highly local. The

BFS typically terminates after inspecting only the immediate neighbors of the difference set.

Substituting the empirical $\bar{h} \approx 1$ into the cost function, the effective complexity becomes:

$$T_{\text{prac}} \approx \sum_{iter=1}^{N/2} O(|\text{DD}|_{iter} \cdot \langle k \rangle) \approx O(N^2 \langle k \rangle_{sparse}). \quad (14)$$

Crucially, while the asymptotic form $O(N^2)$ remains, the constant factor drops from the global edge count E_{tot} to the local degree $\langle k \rangle$. In massive real-world networks where $\langle k \rangle \ll N$, this locality ensures that the algorithm operates on a tiny fraction of the graph per iteration, enabling the near-linear runtime scaling observed in subsequent experiments.

4.5 Illustrative Case Study

To make the CLAP mechanism concrete, we visualize a complete execution trace on a toy duplex network ($N = 9$, $k_1 = k_2 = 4$) in Fig. 3.

Starting from (A), we have $\text{DD}_1 = \{1, 3, 9\}$, $\text{DD}_2 = \{4, 6, 8\}$, $\text{CDS} = \{5\}$ and $\text{CDS} = \{7\}$. CLAP-S then iteratively finds and applies the shortest CLAPs: $\mathcal{P}_1 : 1 \xrightarrow{1} 4$ (C1), $\mathcal{P}_2 : 3 \xrightarrow{1} 2 \xrightarrow{2} 4 \xrightarrow{1} 5 \xrightarrow{2} 6$ (C2), and $\mathcal{P}_3 : 9 \xrightarrow{1} 6 \xrightarrow{2} 7 \xrightarrow{1} 8$ (C3), yielding the terminal driver sets $D_1 = D_2 = \{2, 5, 6, 8\}$. Concretely, after applying $\mathcal{P}_1$, we obtain $(D_1, D_2) = (\{3, 4, 7, 9\}, \{4, 6, 7, 8\})$, and after applying $\mathcal{P}_2$, we obtain $(D_1, D_2) = (\{2, 5, 7, 9\}, \{2, 5, 7, 8\})$.

This example also makes the relay-type restriction in Lemma 1 tangible: relays reached via a layer-1 segment must lie in CMS (e.g., 2 and 5 in $\mathcal{P}_2$, and 6 in $\mathcal{P}_3$), whereas relays reached via a layer-2 segment must lie in CDS (e.g., 4 in $\mathcal{P}_2$ and 7 in $\mathcal{P}_3$). This restriction ensures that each chained exchange is admissible and budget preserving. Intuitively, a relay reached via layer 1 must also be a non-driver in layer 2 in order to serve as the starting node of the next layer-2 admissible segment; hence, it must lie in CMS; similarly, a relay reached via layer 2 must be a driver in both layers (in CDS) to initiate the subsequent layer-1 segment.

5 Experimental Results and Analysis

To comprehensively evaluate the performance of our proposed CLAP-S algorithm, we conducted extensive experiments on both synthetically generated networks and a diverse collection of real-world duplex network systems. This section details the experimental setup, including the datasets, the baseline algorithm used for comparison, and the performance metrics.

5.1 Description of the Network Dataset

In this section, we describe the experimental environment, the diverse datasets employed for validation, and the quantitative metrics used to assess optimization quality and computational efficiency.

5.1.1 *Synthetic Networks*

To systematically investigate the algorithm's performance under controlled conditions, we generated duplex networks using two canonical models: Erdős-Rényi (ER) random graphs [71] and Barabási-Albert (BA) scale-free networks [72]. These models allow us to test CLAP-S across networks with homogeneous and heterogeneous degree distributions [71], [72].

For standard performance comparisons, the number of nodes N is fixed at 1000. We systematically varied the average degree $\langle k \rangle$ from 2.0 to 10.0, enabling the study of networks ranging from sparse to relatively dense topologies. The construction of a duplex network involved first generating a base layer (Layer 1) and then generating a second layer (Layer 2) with a specified Jaccard similarity of their edge sets, allowing us to control the inter-layer structural overlap from low (0.1) to high (0.9). This process yielded duplex ER networks (ER-ER), duplex BA networks (BA-BA), and hybrid duplex networks (ER-BA), providing a robust dataset for evaluating the CLAP-S algorithm.

5.1.2 *Real-World Networks*

To assess the applicability of CLAP-S in practical scenarios, we curated a diverse dataset of 22 empirical duplex networks from the CoMuNe Lab collection [68]. These networks span genetic and neuronal interactions (e.g., gene regulation, connectomes), online social behaviors (e.g., Twitter retweets/mentions), and offline human relationships, covering a wide spectrum of scales from small communities ($N \approx 20$) to massive social platforms ($N \approx 7.5 \times 10^5$). For each system, we constructed a consistent duplex structure by aligning two representative layers on the union of their node sets. Detailed topological statistics, layer semantics, and network properties are provided in **Supplementary Sec. 5**.

5.2 Baseline Algorithms

We benchmark CLAP-S against three baselines:

5.2.1 RSU *(Random Sample & Union)*

This baseline exploits the non-uniqueness of MDSs via stochastic sampling. RSU generates S random maximum matchings for each layer ℓ, denoted as $\mathcal{S}_\ell = \{\text{MDS}_{\ell,1}, \ldots, \text{MDS}_{\ell,S}\}$. It then computes all $S \times S$ pairwise unions and returns the configuration yielding the minimum cardinality: $|\text{UDS}|_{\text{RSU}} = \min_{i,j} |\text{MDS}_{1,i} \cup \text{MDS}_{2,j}|$. We set $S = 20$ based on sensitivity analysis (see **Supplementary Sec. 6**), which balances optimization quality against the quadratic computational overhead.

5.2.2 CLAP-G *(Greedy Segment)*

CLAP-G serves as a fast local-search heuristic. Starting from an initial state, it iteratively searches for and applies the *first found* single admissible segment (a length-1 CLAP fragment) that strictly reduces $|U|$, terminating when no such local move exists. Unlike CLAP-S, it lacks global shortest-path guidance and batched feasibility checks.

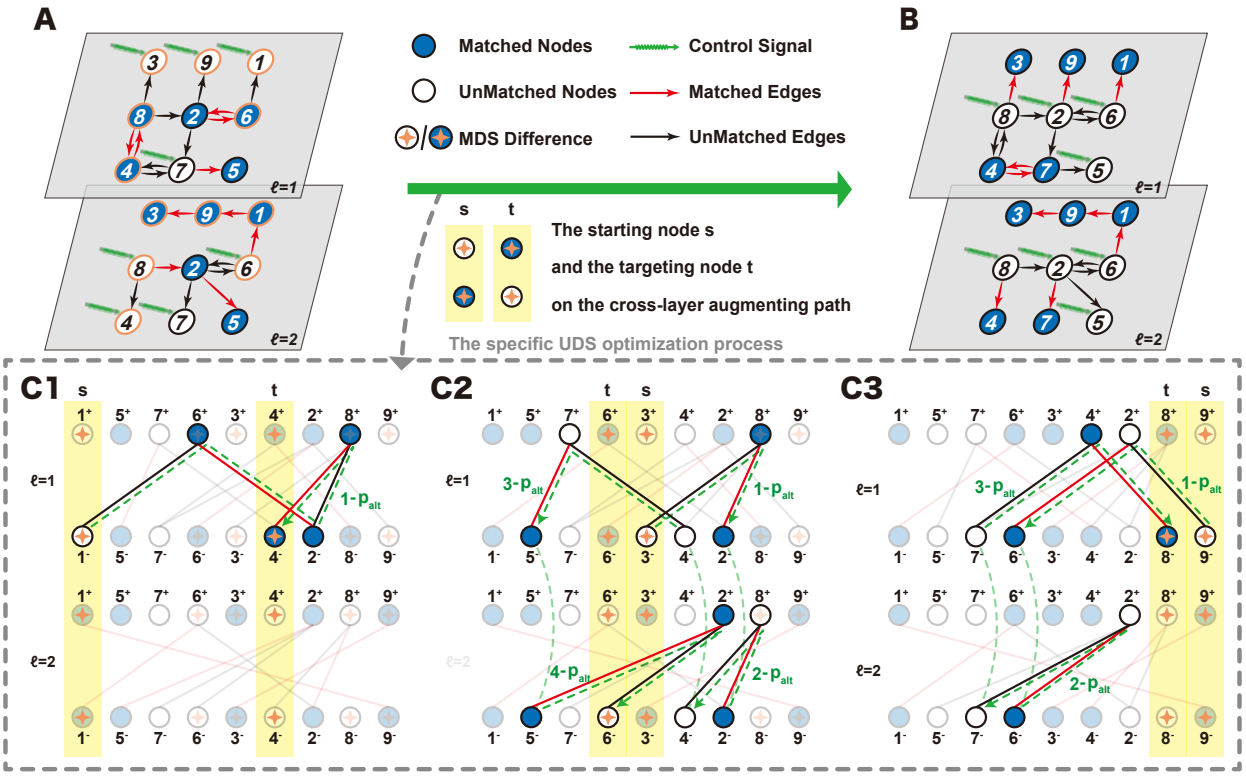

Fig. 3. **Execution Trace of CLAP-S.** (A) **Initial State:** Arbitrary max matchings yield disjoint driver sets. $\mathrm{DD}_1 = \{1, 3, 9\}$, $\mathrm{DD}_2 = \{4, 6, 8\}$. The difference mass is $\Delta = 6$. (C1) **Iteration 1:** The algorithm finds a direct length-1 CLAP ($1 \xrightarrow{1} 4$). Node 1 (L1-Release) and Node 4 (L1-Absorb) are eliminated. Δ drops to 4. (C2) **Iteration 2:** A length-3 CLAP is found ($3 \xrightarrow{1} 2 \xrightarrow{2} 4 \xrightarrow{1} 5 \xrightarrow{2} 6$). Note that relay nodes 2 and 5 are in CMS, and 4 and 5 are in CDS (intermediate state), satisfying Lemma 1. (C3) **Iteration 3:** A final path eliminates the remaining difference. (B) **Optimal State:** The system reaches $\Delta = 0$ with perfect alignment ($D_1 = D_2$). No further CLAPs exist (Stability), certifying optimality.

5.2.3 ILP-EXACT *(Exact Solver)*

We formulate the union minimization as a budget-constrained integer linear program (ILP) and solve it using the CP-SAT solver in OR-Tools [73]. This provides the ground-truth global optimum for assessing solution quality on tractable instances.

5.2.4 *Implementation and Environment*

Our implementation, including all baselines and reproduction scripts, is available at https://github.com/njnklab/CLAP-S_Algorithm. The code is written in Python and relies on `networkx` for graph operations and, optionally, OR-Tools or PuLP for the ILP baseline [73], [74], [75]. All experiments were conducted on a workstation equipped with an Intel® Core™ i9-14900K processor, featuring 24 cores and 32 threads, with a base frequency of 3.2 GHz. The system includes 125 GiB of DDR5 RAM. The operating system was Ubuntu 24.04.2 LTS (codename *noble*) with a 64-bit x86_64 architecture.

5.3 Performance Metrics

We evaluate optimization quality relative to an initial state $\mathcal{I} = (\mathcal{M}_1, \mathcal{M}_2)$, where matchings are selected independently. The primary metric is the **Absolute UDS Reduction** (ΔN_D), quantifying the number of unique drivers saved:

$$\Delta N_D(\mathcal{A} \mid \mathcal{I}) = |\mathrm{UDS}|_{\mathcal{I}} - |\mathrm{UDS}|_{\mathcal{A}}, \tag{15}$$

where $|\mathrm{UDS}|_{\mathcal{A}}$ is the union size achieved by algorithm $\mathcal{A}$.

To verify the advantage of guided search over random sampling, we define the **Optimization Gain** (ΔN_D^{opt}) as the additional reduction achieved by CLAP-S over RSU:

$$\Delta N_D^{opt} = |\mathrm{UDS}|_{\mathrm{RSU}} - |\mathrm{UDS}|_{\text{CLAP-S}}. \tag{16}$$

Finally, the **Relative Optimization Rate** (R_{opt}) measures the percentage improvement over the randomized baseline:

$$R_{\mathrm{opt}} = \frac{\Delta N_D^{opt}}{|\mathrm{UDS}|_{\mathrm{RSU}}} \times 100\%. \tag{17}$$

5.4 Results on Synthetic Networks

We evaluate CLAP-S, its greedy variant CLAP-G, the randomized baseline RSU, and an integer linear programming solver (ILP-EXACT) on synthetic duplex networks with $N = 1000$ nodes. For each parameter setting, we average over 10 independent duplex instances.

5.4.1 *UDS Optimization Count across Network Types and Densities*

Fig. 4 summarizes the results. Two facts stand out. **(i) CLAP-S matches ILP exactly.** The ILP-EXACT and CLAP-S polylines are *perfectly overlapped* in all panels and degree groups, showing that CLAP-S attains the same number of saved drivers as ILP-EXACT on every tested configuration (within plotting precision). **(ii) CLAP-S dominates RSU, and CLAP-G bridges quality and speed.** The box plots are strictly above zero except in the densest regimes, confirming

that CLAP-S consistently saves more drivers than RSU. The greedy CLAP-G line trails CLAP-S/ILP-EXACT in sparse regimes but remains well above RSU, reflecting a favorable quality–speed tradeoff.

Density and topology shape the headroom for optimization. As the average degree $\langle k \rangle$ increases, all methods converge toward smaller improvements because the initial UDS shrinks and cross-layer exchange opportunities diminish. Across topologies, BA–BA exhibits a pronounced peak in the moderately sparse group ($\langle k \rangle \in [4, 5.9]$), where CLAP-S/ILP-EXACT save the most drivers on average; ER–ER steadily declines with density; the hybrid ER–BA falls in between, consistent with its mixed structure.

We hypothesize that the phenomenon of BA-BA arises from the unique interplay between the scale-free degree distribution and the configuration space of maximum matchings. In the ultra-sparse regime ($\langle k \rangle \approx 2$), the abundance of degree-1 nodes and small tree-like components "forces" a large number of driver nodes into specific, non-interchangeable peripheral positions. This topological rigidity leaves little room for CLAP-S to reconfigure matchings for cross-layer alignment. Conversely, in dense regimes ($\langle k \rangle > 10$), the total driver budget k_ℓ itself becomes small, naturally limiting the absolute headroom for union contraction.

The moderately sparse range ($4 \leq \langle k \rangle \leq 6$) represents a "sweet spot" of **structural flexibility**. In scale-free networks, hubs are almost always matched in any maximum matching, while driver nodes are predominantly low-degree peripheral nodes. In this specific density range, the network is connected enough to sustain a high degree of *matching degeneracy*—meaning there exist many distinct maximum matchings that yield the same minimum driver set size. CLAP-S effectively navigates this vast configuration space, utilizing alternating paths to swap peripheral driver nodes until they overlap across layers. In contrast, ER networks, being more homogeneous, possess a more rigid and less degenerate matching structure at similar densities, explaining why their optimization curve remains flatter and declines more steadily.

5.4.2 *Impact of Network Overlap and Density on* CLAP-S *Performance*

To further dissect the performance of our algorithm, we systematically investigated the combined influence of network density (average degree $\langle k \rangle$) and inter-layer structural similarity (network overlap ratio ρ).

Fig. 5 presents the optimization performance versus the overlap ratio ρ. CLAP-S demonstrates remarkable **structural robustness**: ΔN_D remains high across $0.1 \leq \rho \leq 0.9$. Even at 10% overlap, it achieves substantial reductions (approx. 50–70 nodes), far exceeding the RSU baseline. This counter-intuitive "similarity insensitivity" suggests that optimization relies on **matching degeneracy**—the overlap of unmatched-node candidate sets—rather than specific edge identities. Furthermore, in heterogeneous topologies (BA), driver positions are structurally pinned to peripheral nodes, ensuring high overlap potential even when the specific wiring differs significantly.

Expanding from this specific slice, Fig. 6 visualizes the graph optimization gain (ΔN_D^{opt}) and the relative optimization rate (R_{opt}) across the parameter space for both ER-ER and BA-BA duplex networks.

The analysis reveals distinct drivers for absolute and relative performance. The absolute gain ΔN_D^{opt} (Fig. 6, Left) is primarily dictated by **network density**. Gains peak in sparse regimes where the pool of difference nodes is largest. While ER gains decline with density, BA networks sustain high gains over a broader range due to their heterogeneous topology. Conversely, the relative optimization rate R_{opt} (Fig. 6, Right) highlights **search complexity**. In ER networks, R_{opt} rises with both density and overlap ρ, indicating that while optimization opportunities shrink, they become disproportionately harder for random sampling to resolve. CLAP-S excels here by identifying these subtle reconfiguration pathways. BA networks show consistently high R_{opt} ($> 12\%$) across the board, confirming that scale-free complexity inherently challenges stochastic approaches.

In summary, the results reveal a dichotomy between *optimization potential* and *search complexity*. *1) Potential is Governed by Density*: The vertical contours of ΔN_D^{opt} confirm that the **optimization ceiling** is dictated by sparsity, which provides the necessary matching degeneracy and reconfiguration headroom. *2) Difficulty is Governed by Similarity*: The increase of R_{opt} with ρ suggests that structural similarity masks driver conflicts, turning alignment into a "needle-in-a-haystack" problem for stochastic baselines. CLAP-S's path-guided mechanism successfully resolves these hidden conflicts, maximizing its relative advantage as layers become more similar.

5.4.3 *Relationship between Initial MDS Difference and* CLAP-S *Optimization*

To characterize the fundamental limits of duplex control coordination, we analyze the relationship between the initial structural discord—quantified by the size of the initial difference driver sets $|\text{DD}_1 \cup \text{DD}_2|$—and the resulting absolute reduction ΔN_D achieved by CLAP-S. As visualized in Fig. 7, a strong positive correlation exists for both network models, confirming that the initial level of disagreement between layers is the primary source of optimization potential.

We introduce a theoretical efficiency frontier with slope $k = 0.5$ (dotted line). This slope derives from Proposition 1: since budgets are fixed, saving one driver node requires eliminating exactly two difference nodes ($\Delta N_D = \Delta/2$). In an ideal "perfect nesting" scenario where driver sets fully overlap ($\Delta_{\text{final}} \approx 0$), the reduction would be exactly half the initial difference. Thus, proximity to this line measures how efficiently the network topology permits conflict resolution.

The results reveal a topological dichotomy. ER networks (Top) exhibit a tight linear correlation ($R^2 = 0.81$) closely tracking the efficiency frontier, suggesting that homogeneous matching degeneracy allows CLAP-S to consistently resolve $\approx 90\%$ of conflicts. In contrast, BA networks (Bottom) show higher dispersion ($R^2 = 0.55$) and a larger gap from the frontier. This vertical deviation quantifies irreducible structural constraints: in scale-free topologies, hubs or their neighbors are often "pinned" to specific match-

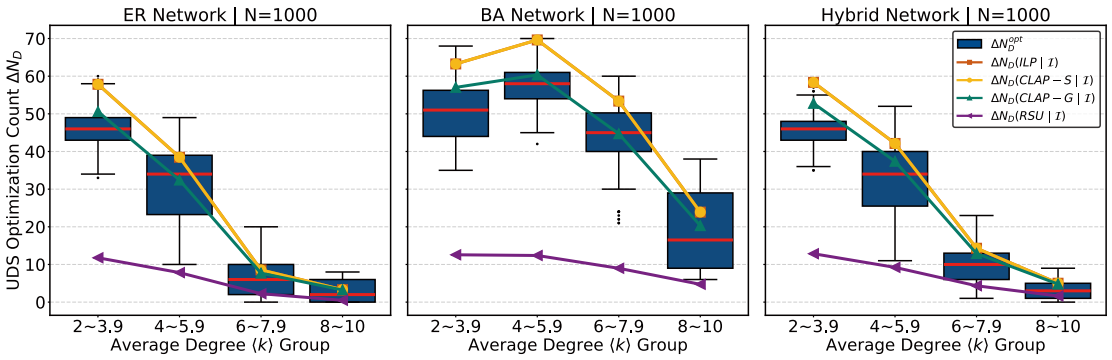

Fig. 4. UDS Optimization Count on synthetic duplex networks ($N = 1000$). Blue box: $\Delta N_D^{\text{opt}} = |UDS|_{\text{RSU}} - |UDS|_{\text{CLAP-S}}$ (additional drivers saved by CLAP-S over RSU). Orange squares: $\Delta N_D(\text{ILP-EXACT} \mid \mathcal{I})$. Yellow circles: $\Delta N_D(\text{CLAP-S} \mid \mathcal{I})$ (perfectly overlapped with ILP). Green triangles: $\Delta N_D(\text{CLAP-G} \mid \mathcal{I})$. Purple arrows: $\Delta N_D(\text{RSU} \mid \mathcal{I})$. Panels: ER–ER, BA–BA, and ER–BA (Hybrid), grouped by average degree $\langle k \rangle$. Higher values indicate more drivers saved relative to the initial state.

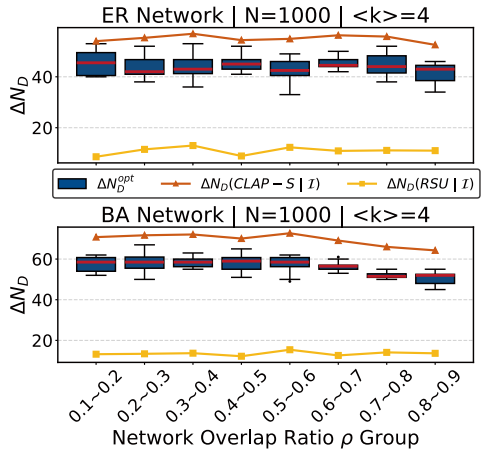

Fig. 5. Cross-sectional analysis of optimization performance versus network overlap ratio ρ at $\langle k \rangle = 4$. Orange lines: CLAP-S absolute reduction; Yellow lines: RSU baseline; Blue box plots: net optimization gain ΔN_D^{opt}.

ing states, creating bottlenecks that even an optimal solver cannot bypass.

Finally, network density (point size) governs the optimization headroom. Sparser networks (small points) cluster in the upper-right high-gain region, while denser networks naturally contract toward the origin as MDS sizes and exchange opportunities diminish. Overall, initial structural discord sets the optimization potential, while topology determines the efficiency of its realization.

5.5 Results on Real-World Networks

We validate CLAP-S on 22 empirical duplex networks ($20 \leq N \leq 7 \times 10^5$) spanning biological, social, and technological domains (Table 3). Performance is evaluated via wall-clock time and peak algorithmic memory, measured using Python's `tracemalloc` module to isolate the dynamic algorithmic footprint from static data-loading overheads.

The data points in Fig. 8 exhibit a strong positive correlation ($R^2 = 0.79$) only when visualized on a log-log scale. This linear trend on the logarithmic plane implies a power-law scaling relationship between the optimization gain and the initial structural discord: $\Delta N_D \propto (|\text{DD}_1 \cup \text{DD}_2|)^{\alpha}$. The observed slope of $\alpha \approx 1$ indicates a property of scale invariance: the algorithm maintains a consistent redundancy recovery rate regardless of the network size. Whether in a small metabolic pathway or a massive retweet network, CLAP-S effectively identifies and resolves a stable proportion of the cross-layer conflicts, converting them into shared actuator locations.

The necessity of the log-log transformation also reveals a fundamental topological constraint inherent to real-world systems: *structural rigidity*. Unlike synthetic random graphs where driver nodes are often interchangeable, real-world networks are characterized by high driver occupancy ratios—frequently, the union of driver sets covers over 80–90% of the entire network (see **Supplementary Sec. 10** for a quantitative comparison of driver occupancy between synthetic and empirical datasets). If visualized on a linear scale, the optimization would appear proportionally small relative to the total system size. This phenomenon stems from domain-specific topological features: for example, in social networks, the *Matthew effect* creates extreme hierarchies wherein a vast number of peripheral nodes have low or zero in-degree; similarly, in gene regulatory networks, functional modularity creates localized, asymmetric control dependencies. These structures effectively lock a majority of nodes into unavoidable driver roles, severely compressing the maneuvering room for optimization. Consequently, CLAP-S yields the most dramatic relative reductions in networks with moderate density and lower driver occupancy; yet it remains crucial for rigid networks by identifying the few but non-trivial coordination opportunities that stochastic sampling misses.

Computationally, CLAP-S matches the exact ILP-EXACT optimum on all feasible instances ($N \leq 10^4$). On massive networks ($N > 3 \times 10^5$), while BFS frontier expansion increases runtime, CLAP-S maintains superior memory efficiency. For example, on *NBAFinals*, it consumes $\approx$ 266 MiB versus RSU's $>$ 2.2 GiB. This order-of-magnitude advantage arises because RSU stores large candidate pools for sampling, whereas CLAP-S operates via lightweight in-place modifications, ensuring scalability.

For massive regimes, the greedy variant CLAP-G offers a pragmatic alternative, delivering near-optimal reductions orders of magnitude faster. Finally, the consistently low av-

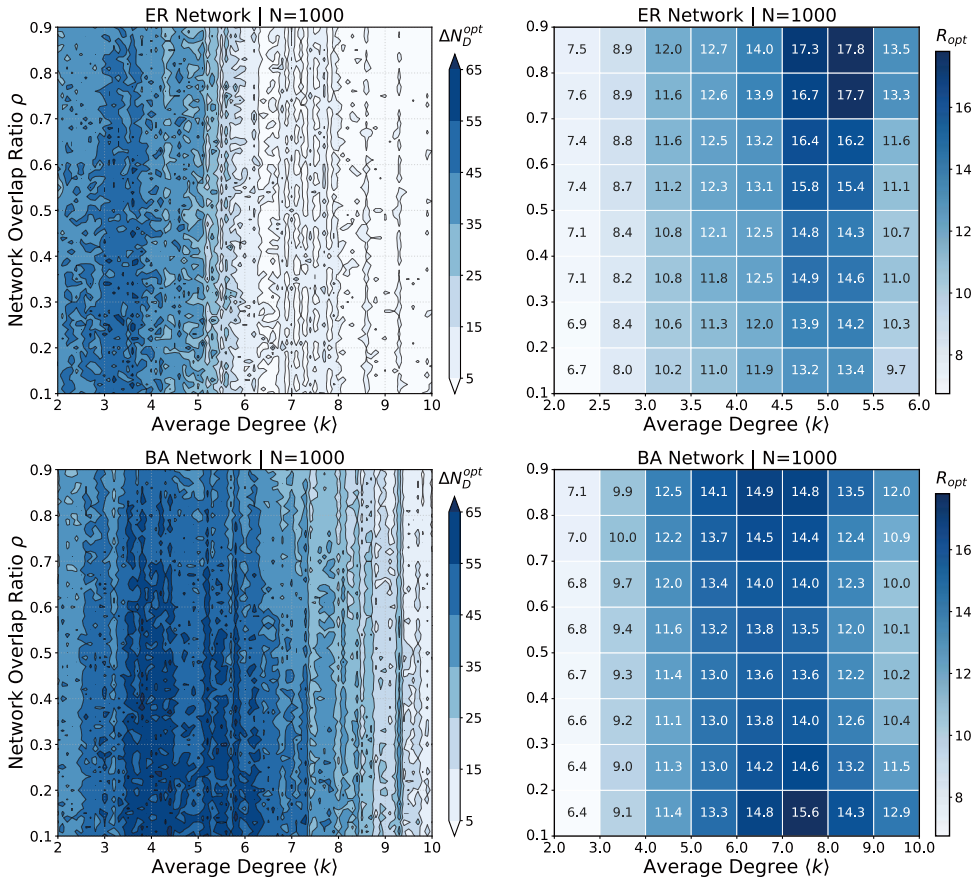

Fig. 6. Impact of network overlap (ρ) and average degree ($\langle k \rangle$) on CLAP-S's performance for ER-ER (top) and BA-BA (bottom) duplex networks ($N = 1000$). Left panels: Contour plots of the graph optimization gain, ΔN_D^{opt}. Right panels: Heatmaps of the relative optimization rate, R_{opt} (in %).

TABLE 3
Detailed performance comparison of CLAP-S and other algorithms on real-world networks. $|U|_0$ is the initial UDS size. $|U|_{\text{CLAPS}}$, $|U|_{\text{RSU}}$, $|U|_{\text{CLAPG}}$, and $|U|_{\text{ILP}}$ are the final sizes after running the respective algorithms. $\overline{h}$ is the average CLAP length found by CLAP-S. t is the execution time in seconds. m is the peak algorithmic memory usage in MiB, measured as the peak allocation delta via `tracemalloc`. ILP was not run for networks with more than 10000 nodes, also marked with '–'.

Network Name	$\lvert U\rvert_0$	$\lvert U\rvert_{\text{CLAP-S}}$	$\lvert U\rvert_{\text{RSU}}$	$\lvert U\rvert_{\text{CLAP-G}}$	$\lvert U\rvert_{\text{ILP}}$	$\overline{h}$	$t_{\text{CLAP-S}}$	t_{RSU}	$t_{\text{CLAP-G}}$	t_{ILP}	$m_{\text{CLAP-S}}$	m_{RSU}	$m_{\text{CLAP-G}}$	m_{ILP}
Arabidopsis	6598	6453	6577	6469	6453	1.193	0.239	11.633	0.175	6.087	2.764	17.691	3.355	13.833
Celegans	3149	3114	3138	3118	3114	1.114	0.018	4.118	0.016	1.308	1.098	8.649	1.395	5.738
Drosophila	7385	7000	7342	7028	7000	1.135	1.472	16.602	0.725	12.718	7.977	17.706	8.960	21.904
HumanHIV1	986	977	984	978	977	1.556	0.025	0.709	0.001	0.272	0.492	2.186	0.290	1.722
SacchPomb	2362	2267	2353	2276	2267	1.211	0.294	4.014	0.071	1.913	1.538	5.912	1.159	6.042
Rattus	2448	2381	2438	2389	2381	1.179	0.108	3.401	0.036	0.937	1.166	8.357	1.135	4.228
CelegansConnectome	62	55	60	58	55	1.429	0.013	0.567	0.001	0.187	0.093	0.188	0.037	1.509
YeastLandscape	3664	2757	3618	2757	2757	1.021	3.491	8.291	3.862	39.091	182.703	8.647	182.412	73.391
Cannes	421156	419243	421058	419300	-	1.043	1139.378	901.057	274.907	-	142.661	1110.335	163.821	-
MLKing	321954	321726	321945	321730	-	1.022	177.033	520.637	15.714	-	63.967	1037.008	90.999	-
MoscowAthletics	86374	86162	86357	86165	-	1.033	26.905	135.560	4.693	-	28.382	273.326	35.916	-
NYClimate	99465	99154	99426	99164	-	1.032	38.172	176.291	7.031	-	34.480	277.575	39.658	-
NBAFinals	741862	741128	741795	741142	-	1.025	1244.294	1381.163	137.649	-	266.608	2219.370	307.209	-
Sanremo	53819	53491	53798	53507	-	1.061	8.974	116.668	3.341	-	18.595	138.853	22.422	-
UCLFinal	672121	671441	672082	671465	-	1.040	801.042	1212.377	114.600	-	218.340	2106.109	275.159	-
GravitationalWaves	357106	356772	357081	356776	-	1.015	240.769	591.890	27.302	-	133.787	1109.630	154.658	-
KrackhardtHighTech	2	2	2	2	2	0	0.002	0.112	< 0.001	0.037	0.022	0.036	0.007	0.409
KrackhardtHighTech	16	16	16	16	16	0	< 0.001	0.058	< 0.001	0.033	0.007	0.036	0.009	0.345
LazegaLawFirm	6	6	6	6	6	0	< 0.001	0.400	< 0.001	0.086	0.016	0.052	0.016	0.874
LazegaLawFirm	7	6	6	6	6	1	< 0.001	0.249	< 0.001	0.094	0.019	0.051	0.018	0.942
PhysiciansInnovation	125	120	121	120	120	1	0.001	0.252	0.001	0.102	0.071	0.299	0.064	0.862
PhysiciansInnovation	104	99	101	99	99	1	0.002	0.782	< 0.001	0.104	0.063	0.219	0.042	0.893

erage path length ($\bar{h} \approx 1.2$) confirms that despite structural rigidity, solvable multi-layer conflicts are predominantly local, allowing our framework to resolve global redundancies through short-range, high-impact reconfigurations.

6 Discussion

In this work, we addressed the problem of identifying a concise, unified set of driver nodes for duplex networks under the practical engineering constraint of fixed, layer-specific driver budgets. We framed this challenge as a budget-

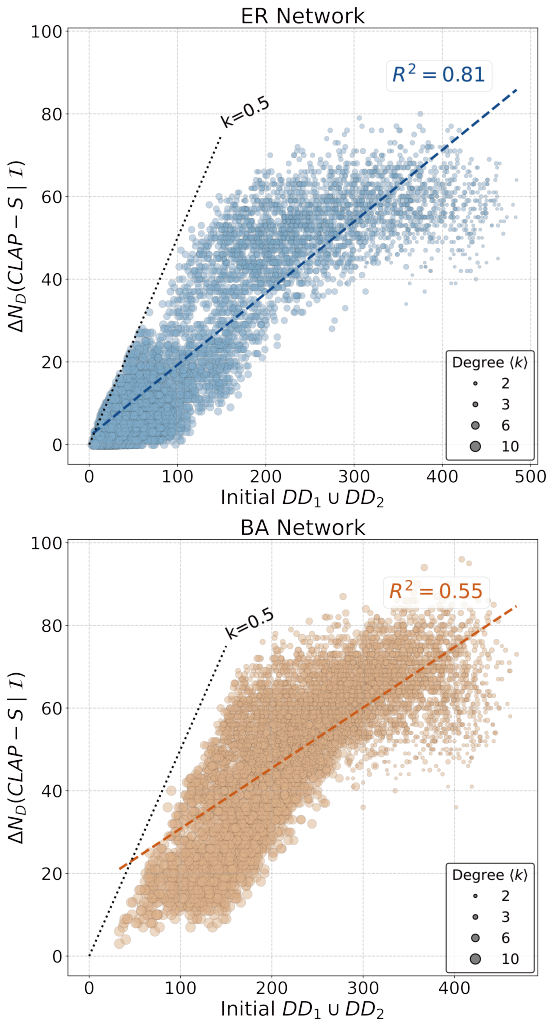

Fig. 7. Relationship between the initial difference driver set size ($|\mathrm{DD}_1 \cup \mathrm{DD}_2|$) and the number of nodes optimized by CLAP-S ($\Delta N_D(\text{CLAP-S} \mid \mathcal{I})$). Each point represents a synthetic duplex network with $N = 1000$. Point size indicates average degree $\langle k \rangle$. The dashed lines are linear regression fits for ER networks ($R^2 = 0.81$) and BA networks ($R^2 = 0.54$). The dotted line represents a high-efficiency conversion slope of $k = 0.5$.

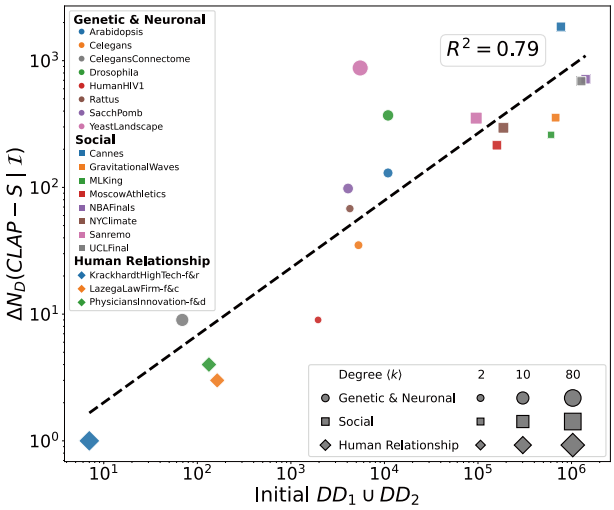

Fig. 8. Optimization performance of CLAP-S on real-world duplex networks (log–log scale). Each marker is a network; color encodes domain (Genetic & Neuronal, Social, Human Relationship) and marker size encodes average degree $\langle k \rangle$. The x-axis is the initial difference size $|\mathrm{DD}_1 \cup \mathrm{DD}_2|$; the y-axis is the number of drivers saved by CLAP-S, $\Delta N_D(\text{CLAP-S} \mid \mathcal{I})$. The dashed line is a least-squares fit with coefficient of determination $R^2 = 0.79$, indicating a strong, approximately linear scaling on the log–log plane. Duplexes with $|\mathrm{DD}_1 \cup \mathrm{DD}_2| = 0$ are not shown since they offer no optimization space.

preserving union contraction problem and introduced a novel, path-based framework to solve it. Our main contribution is the CLAP-S algorithm, a deterministic, polynomial-time procedure that guaranties monotonic improvement and terminates at a provably irreducible state.

6.1 Summary of Contributions and Positioning

The core contribution of CLAP-S is the CLAP-Stability Theorem, which establishes that a locally stable state (no augmenting paths) corresponds to a global minimum over the feasible set. This provides a rigorous optimality certificate often missing in heuristic network control approaches. Empirically, CLAP-S matches the solution quality of exact ILP solvers while outperforming random sampling baselines by orders of magnitude in speed.

It is crucial to position our work within the broader context of combinatorial optimization. **Distinction from Related Problems:** Unlike general matching reconfiguration, which is often PSPACE-complete [62], [76], or the minimization of driver-plus-FVS unions, which is NP-hard, our problem is tractable. The tractability stems from the fixed-budget constraint, which confines the search to the intersection of two matroid-like structures (layer-wise maximum matchings), allowing efficient navigation via augmenting paths.

6.2 Implications and Limitations

The concept of a CLAP-stable state has significant practical implications. It provides an operational criterion to assess whether a given configuration of driver nodes is optimally aligned. For a systems engineer, the absence of a CLAP is a clear signal that no further UDS contraction can be achieved without altering the control budget of at least one layer. This aligns with real-world scenarios where layer-specific performance guaranties (represented by the driver budgets) are non-negotiable.

Our notion of "simultaneous control" is intentionally defined under an *aligned-node, uncoupled duplex* abstraction: (i) both layers share the same node set (the same physical units), (ii) inter-layer edges are absent, and (iii) per-layer controllability is enforced under fixed budgets k_1, k_2. These assumptions match engineering settings where each layer represents an independent mode/architecture of the *same* system, and actuators (driver nodes) can be physically shared across modes but must satisfy layer-specific requirements. When explicit inter-layer coupling edges are introduced (e.g., supra-adjacency models with cross-layer links between counterpart nodes), the dynamics no longer decomposes into two independent structural controllability problems, and the meaning of per-layer minimal budgets becomes model-dependent. In that coupled setting, the natural formulation is structural controllability of the *supra-system* with a shared actuation pattern, which constitutes

a different problem class and requires new theory beyond the present budget-preserving duplex framework.

The primary limitation of our framework is structural rather than algorithmic. The final contracted UDS size is determined by the intrinsic properties of the network layers. While CLAP-S is guaranteed to find the minimum possible union within the fixed-budget regime, this minimum may not necessarily reach the theoretical lower bound of $\max\{k_1, k_2\}$. Such a bound is only attainable if the network topology permits a perfect nesting of one driver set within another, a condition that is not always met. Our algorithm correctly identifies the true, achievable minimum, whatever it may be.

Our optimality guarantee is stated for the practically relevant regime where each layer individually uses the minimum actuation budget implied by maximum matching, i.e., (k_1, k_2) fixed and budget-preserving moves only. If a practitioner allows relaxed budgets $\tilde{k}_\ell \geq k_\ell$, our solution remains immediately feasible (hence an admissible upper bound on the union under relaxed budgets), but the *globally* minimum union under $\tilde{k}_\ell$ may become strictly smaller and would require searching a larger feasibility domain beyond maximum matchings. Therefore, we do not treat relaxed budgets as a limitation of the proposed algorithm; rather, it defines a different optimization problem whose study is complementary to, but distinct from, the MinUDS formulation addressed here.

Another structural limitation is the potential for relay node scarcity. The existence of CLAPs depends on a sufficient supply of nodes in the CDS and CMS sets to serve as relays. In networks where these sets are small, CLAPs may be non-existent, leading to a state that is CLAP-stable but may still appear suboptimal from a global, budget-violating perspective.

6.3 Future Work

This work opens several promising avenues for future research. A natural extension would be to relax the fixed-budget constraint and explore controlled trade-offs. For instance, one could investigate algorithms that allow a small budget increase in one layer if it leads to a disproportionately large decrease in the total UDS size, formulating the problem in a multi-objective optimization framework.

Extending the CLAP framework to multiplex networks with more than two layers is another important direction. One natural generalization is to define budgets $\{k_\ell\}_{\ell=1}^{L}$ and minimize the union size $|\cup_{\ell=1}^{L} \mathcal{D}_\ell|$ over $\prod_{\ell=1}^{L} \mathbb{M}_\ell(k_\ell)$. A practical algorithmic route is a *pairwise coordinate-descent* strategy: iteratively apply duplex CLAP-S updates to selected layer pairs (ℓ_i, ℓ_j) while keeping the remaining layers fixed, which monotonically improves the global union and provides a scalable baseline. A more ambitious route is to define *multi-layer CLAPs* as alternating segments that traverse an ordered sequence of layers, allowing a single operation to reconcile discrepancies across multiple layers at once. The key challenge is that the implicit state space grows with L, and the path structure is no longer captured by a single duplex difference mass Δ; maintaining polynomial-time guarantees would likely require new certificates and carefully designed local moves.

Finally, incorporating explicit inter-layer coupling edges is a meaningful but nontrivial extension. Although coupled multiplexes can be represented as a supra-graph, the control problem then becomes structural controllability of the coupled supra-system with a shared actuation pattern, and the notion of per-layer minimal driver budgets is no longer directly applicable. Developing CLAP-like local transformations and optimality certificates for such coupled settings would require redefining the feasible set and the objective (e.g., actuator sharing under supra-dynamics), which we leave for future work.

APPENDIX

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62176129; the NSFC–Jiangsu Joint Fund under Grant U24A20701; and the Hong Kong RGC Strategic Target Grant (Grant No. STG1/M-501/23-N).

REFERENCES

[1] M. Newman, *Networks*. Oxford University Press, Jul. 2018. [Online]. Available: https://doi.org/10.1093/oso/9780198805090.001.0001

[2] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, pp. 268–276, Mar. 2001, publisher: Springer Science and Business Media LLC. [Online]. Available: http://dx.doi.org/10.1038/35065725

[3] J. Ruths and D. Ruths, "Control Profiles of Complex Networks," *Science*, vol. 343, no. 6177, pp. 1373–1376, Mar. 2014, publisher: American Association for the Advancement of Science (AAAS). [Online]. Available: http://dx.doi.org/10.1126/science.1242063

[4] S. Gu, F. Pasqualetti, M. Cieslak, Q. K. Telesford, A. B. Yu, A. E. Kahn, J. D. Medaglia, J. M. Vettel, M. B. Miller, S. T. Grafton, and D. S. Bassett, "Controllability of structural brain networks," *Nature Communications*, vol. 6, no. 1, Oct. 2015, publisher: Springer Science and Business Media LLC. [Online]. Available: http://dx.doi.org/10.1038/ncomms9414

[5] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, May 2011, publisher: Springer Science and Business Media LLC. [Online]. Available: http://dx.doi.org/10.1038/nature10011

[6] Ching-Tai Lin, "Structural controllability," *IEEE Transactions on Automatic Control*, vol. 19, no. 3, pp. 201–208, Jun. 1974, publisher: Institute of Electrical and Electronics Engineers (IEEE). [Online]. Available: http://dx.doi.org/10.1109/tac.1974.1100557

[7] Y.-Y. Liu and A.-L. Barabási, "Control principles of complex systems," *Reviews of Modern Physics*, vol. 88, no. 3, p. 035006, 2016.

[8] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 40–52, 2014.

[9] A. Olshevsky, "Minimal controllability problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, 2014.

[10] S. Pequito, S. Kar, and A. P. Aguiar, "A framework for structural input/output and control configuration selection in large-scale systems," *arXiv*, 2013, https://arxiv.org/abs/1309.5868.

[11] X. Zhang, J. Han, and W. Zhang, "An efficient algorithm for finding all possible input nodes for controlling complex networks," *Scientific Reports*, vol. 7, no. 1, Sep. 2017, publisher: Springer Science and Business Media LLC. [Online]. Available: http://dx.doi.org/10.1038/s41598-017-10744-w

[12] T. Jia and A.-L. Barabási, "Control capacity and a random sampling approach in exploring controllability of complex networks," *Scientific Reports*, vol. 3, p. 2354, 2013. [Online]. Available: https://www.nature.com/articles/srep02354

[13] S. Jia, Y. Xi, D. Li, and H. Shao, "Finding complete minimum driver node set with guaranteed control capacity," *Neurocomputing*, vol. 500, pp. 949–964, Aug. 2022, publisher: Elsevier BV. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2022.05.095

[14] X. Zhang, "Altering Indispensable Proteins in Controlling Directed Human Protein Interaction Network," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 2074–2078, Nov. 2018, publisher: Institute of Electrical and Electronics Engineers (IEEE). [Online]. Available: http://dx.doi.org/10.1109/tcbb.2018.2796572

[15] X. Piao, T. Lv, X. Zhang, and H. Ma, "Strategy for community control of complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 421, pp. 98–108, Mar. 2015, publisher: Elsevier BV. [Online]. Available: http://dx.doi.org/10.1016/j.physa.2014.10.081

[16] L. C. G. Lebon, F. Lo Iudice, and C. Altafini, "On controllability of temporal networks," *European Journal of Control*, vol. 75, p. 101046, 2024.

[17] C. Pan, X. Zhang, H. Zheng, Y. Zhang, Z. Su, C. Zhang, and W. Zhang, "Adaptive Control of Dynamic Networks," *IEEE Transactions on Network Science and Engineering*, pp. 1–15, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/11150536/

[18] X. Zhang and Q. Li, "Altering control modes of complex networks based on edge removal," *Physica A: Statistical Mechanics and its Applications*, vol. 516, pp. 185–193, Feb. 2019, publisher: Elsevier BV. [Online]. Available: http://dx.doi.org/10.1016/j.physa.2018.09.146

[19] X. Zhang, Y. Zhu, and Y. Zhao, "Altering control modes of complex networks by reversing edges," *Physica A: Statistical Mechanics and its Applications*, vol. 561, p. 125249, Jan. 2021, publisher: Elsevier BV. [Online]. Available: http://dx.doi.org/10.1016/j.physa.2020.125249

[20] S. S. A. Darbandi, A. Fornito, and A. Ghasemi, "The impact of input node placement in the controllability of brain networks," 2023. [Online]. Available: https://arxiv.org/abs/2308.00160

[21] X. Zhang, C. Pan, X. Wei, M. Yu, S. Liu, J. An, J. Yang, B. Wei, W. Hao, Y. Yao, Y. Zhu, and W. Zhang, "Cancer-keeper genes as therapeutic targets," *iScience*, vol. 26, no. 8, p. 107296, Aug. 2023, publisher: Elsevier BV. [Online]. Available: http://dx.doi.org/10.1016/j.isci.2023.107296

[22] C. Dimulescu, S. Gareayaghi, F. Kamp, S. Fromm, K. Obermayer, and C. Metzner, "Structural Differences Between Healthy Subjects and Patients With Schizophrenia or Schizoaffective Disorder: A Graph and Control Theoretical Perspective," *Frontiers in Psychiatry*, vol. 12, Jun. 2021, publisher: Frontiers Media SA. [Online]. Available: http://dx.doi.org/10.3389/fpsyt.2021.669783

[23] A. Sadaf, L. Mathieson, P. Bródka, and K. Musial, "A bridge between influence models and control methods," *Applied Network Science*, vol. 9, no. 1, Jul. 2024, publisher: Springer Science and Business Media LLC. [Online]. Available: http://dx.doi.org/10.1007/s41109-024-00647-x

[24] M. Kivela, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of Complex Networks*, vol. 2, no. 3, pp. 203–271, Jul. 2014, publisher: Oxford University Press (OUP). [Online]. Available: http://dx.doi.org/10.1093/comnet/cnu016

[25] S. Boccaletti, G. Bianconi, R. Criado, C. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin, "The structure and dynamics of multilayer networks," *Physics Reports*, vol. 544, no. 1, pp. 1–122, Nov. 2014, publisher: Elsevier BV. [Online]. Available: http://dx.doi.org/10.1016/j.physrep.2014.07.001

[26] M. De Domenico, A. Solé-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M. A. Porter, S. Gómez, and A. Arenas, "Mathematical formulation of multilayer networks," *Physical Review X*, vol. 3, no. 4, p. 041022, 2013.

[27] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, pp. 1025–1028, 2010.

[28] J. Gao, S. V. Buldyrev, H. E. Stanley, and S. Havlin, "Robustness of a network of networks," *Nature Physics*, vol. 8, pp. 40–48, 2012.

[29] C. Yuan, Z.-Y. Qian, J. Zhou, S.-M. Chen, and S. Nie, "Structural characteristics in network control of molecular multiplex networks," *PLOS ONE*, vol. 18, no. 3, p. e0283768, Mar. 2023, publisher: Public Library of Science (PLoS). [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0283768

[30] M. Szell, R. Lambiotte, and S. Thurner, "Multirelational organization of large-scale social networks," *Proceedings of the National Academy of Sciences*, vol. 107, no. 31, pp. 13 636–13 641, 2010.

[31] G. Menichetti, L. Dall'Asta, and G. Bianconi, "Control of Multilayer Networks," *Scientific Reports*, vol. 6, no. 1, Feb. 2016, publisher: Springer Science and Business Media LLC. [Online]. Available: http://dx.doi.org/10.1038/srep20706

[32] M. Pósfai, J. Gao, S. P. Cornelius, A.-L. Barabási, and R. M. D'Souza, "Controllability of multiplex, multi-time-scale networks," *Physical Review E*, vol. 94, no. 3, p. 032316, 2016.

[33] L. Jiang, L. Tang, and J. Lü, "Controllability of multilayer networks," *Asian Journal of Control*, vol. 24, no. 4, pp. 1517–1527, 2022.

[34] L. Wang, Z. Li, G. Guo, and Z. Kong, "Target controllability of multi-layer networks with high-dimensional nodes," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 9, pp. 1999–2010, 2024.

[35] H. Wang, P. Zhai, S. Ye, P. Ren, and C. Wei, "Controllability of multi-relational networks with heterogeneous dynamical nodes," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 12, pp. 2476–2486, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10625594

[36] W. Someya, F. Morone, and H. A. Makse, "Target control of linear directed networks based on the path-cover problem," *Scientific Reports*, vol. 14, p. 14555, 2024.

[37] Y. Zhang, A. Garas, and F. Schweitzer, "Value of peripheral nodes in controlling multilayer networks," *Physical Review E*, vol. 93, no. 1, p. 012309, 2016.

[38] Q. Li, Q. Liu, M. Medo, Z. Yang, and L. Lü, "Controllability of multilayer networks through single-layer control," 2024. [Online]. Available: https://arxiv.org/abs/2406.06910

[39] V. Srivastava, S. A. Marvel, V. Katewa, F. Pasqualetti, and M. Egerstedt, "Structural underpinnings of control in multiplex networks," 2021. [Online]. Available: https://arxiv.org/abs/2103.08757

[40] J. A. Bernal, N. Alon, M. A. Porter, J.-C. Delvenne, and R. Lambiotte, "Spreading processes on multilayer networks," *Physics Reports*, vol. 829, pp. 1–64, 2020.

[41] J. C. Nacher and T. Akutsu, "Finding and analysing the minimum set of driver nodes required to control multilayer networks," *Scientific Reports*, vol. 9, no. 1, p. 5769, 2019.

[42] H.-T. Zheng, J.-P. Li, Q.-H. Zhang, C.-X. Ren, and G.-R. Liu, "Controllability of biological networks: A survey of recent advances and challenges," *BMC Systems Biology*, vol. 13, no. 1, p. 39, 2019.

[43] H. Han, J.-W. Cho, S. Lee, A. Yun, H. Kim, D. Bae, S. Yang, C. Y. Kim, M. Lee, E. Kim, S. Lee, B. Kang, D. Jeong, Y. Kim, H.-N. Jeon, H. Jung, S. Nam, M. Chung, J.-H. Kim, and I. Lee, "Trrust v2: an expanded reference database of human and mouse transcriptional regulatory interactions," *Nucleic Acids Research*, vol. 46, no. D1, pp. D380–D386, 2018.

[44] D. Marbach, J. C. Costello, R. Küffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, M. Kellis, J. J. Collins, G. Stolovitzky *et al.*, "Wisdom of crowds for robust gene network inference," *Nature Methods*, vol. 9, no. 8, pp. 796–804, 2012.

[45] I. Aldasoro and I. Alves, "Multiplex interbank networks and systemic importance – an application to european data," Bank for International Settlements (BIS), BIS Working Papers 603, Jan 2017.

[46] S. Vitali, J. B. Glattfelder, and S. Battiston, "The network of global corporate control," *PLOS ONE*, vol. 6, no. 10, p. e25995, 2011.

[47] S. A. Myers, A. Sharma, P. Gupta, and J. Lin, "Information network or social network? the structure of the twitter follow graph," in *Proceedings of the 23rd International Conference on World Wide Web (WWW '14 Companion)*, 2014.

[48] J. C. Nacher, M. Ishitsuka, S. Miyazaki, and T. Akutsu, "Finding and analysing the minimum set of driver nodes required to control multilayer networks," *Scientific Reports*, vol. 9, no. 1, Jan. 2019, publisher: Springer Science and Business Media LLC. [Online]. Available: http://dx.doi.org/10.1038/s41598-018-37046-z

[49] B. Wang, X. Ma, C. Wang, M. Zhang, Q. Gong, and L. Gao, "Conserved control path in multilayer networks," *Entropy*, vol. 24, no. 7, p. 979, 2022.

[50] X. Zhang, T. Lv, X. Yang, and B. Zhang, "Structural Controllability of Complex Networks Based on Preferential Matching," *PLoS ONE*, vol. 9, no. 11, p. e112039, Nov. 2014, publisher: Public Library of Science (PLoS). [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0112039

[51] X. Zhang, T. Lv, and Y. Pu, "Input graph: the hidden geometry in controlling complex networks," *Scientific Reports*, vol. 6, no. 1, Nov. 2016, publisher: Springer Science and Business Media LLC. [Online]. Available: http://dx.doi.org/10.1038/srep38209

[52] I. Klickstein and F. Sorrentino, "Controlling network ensembles," *Nature Communications*, vol. 12, p. 1884, 2021.

[53] S. Alizadeh, M. Pósfai, and A. Ghasemi, "Input node placement restricting the longest control chain in controllability of complex networks," *Scientific Reports*, vol. 13, p. 3752, 2023.

[54] S. S. Alizadeh Darbandi, A. Fornito, and A. Ghasemi, "The impact of input node placement in the controllability of structural brain networks," *Scientific Reports*, vol. 14, p. 57181, 2024.

[55] W. Zheng, D. Wang, and X. Zou, "Control of multilayer biological networks and applied to target identification of complex diseases," *BMC Bioinformatics*, vol. 20, no. 1, May 2019, publisher: Springer Science and Business Media LLC. [Online]. Available: http://dx.doi.org/10.1186/s12859-019-2841-2

[56] Y. Sun, Y. Pei, H. Zhang, S. Wang, and A. Zeng, "Robust strong structural controllability of complex power systems," *Frontiers in Energy Research*, vol. 10, p. 913893, 2022.

[57] Y. Li, Y. Ge, Z. Fang, W. Zhang, and X. Jin, "Controllability evaluation of complex networks in cyber–physical power systems via critical nodes and edges," *International Journal of Electrical Power & Energy Systems*, vol. 155, p. 109625, 2024.

[58] M. Rinaldi *et al.*, "Controllability of transportation networks," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 152–168, 2018.

[59] S. S. Mousavi and A. Kouvelas, "Structural controllability of highway networks," Swiss Transport Research Conference (STRC) Report, 2020, https://www.strc.ch/2020/Mousavi_EtAl.pdf.

[60] C. Berge, "TWO THEOREMS IN GRAPH THEORY," *Proceedings of the National Academy of Sciences*, vol. 43, no. 9, pp. 842–844, Sep. 1957, publisher: Proceedings of the National Academy of Sciences. [Online]. Available: http://dx.doi.org/10.1073/pnas.43.9.842

[61] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matching in bipartite graphs," *SIAM Journal on Computing*, vol. 2, no. 4, pp. 225–231, 1973.

[62] M. Bonamy, N. Bousquet, M. Heinrich, T. Ito, Y. Kobayashi, A. Mary, M. Mühlenthaler, and K. Wasa, "The perfect matching reconfiguration problem," in *Proceedings of MFCS 2019*, ser. LIPIcs, vol. 138. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019, pp. 80:1–80:14.

[63] R. E. Kalman, "Mathematical Description of Linear Dynamical Systems," *Journal of the Society for Industrial and Applied Mathematics Series A Control*, vol. 1, no. 2, pp. 152–192, Jan. 1963. [Online]. Available: http://dx.doi.org/10.1137/0301010

[64] J.-M. Dion, C. Commault, and J. van der Woude, "Generic properties and control of linear structured systems: A survey," *Automatica*, vol. 39, no. 7, pp. 1125–1144, 2003.

[65] L. Lovász and M. D. Plummer, *Matching Theory*, ser. Annals of Discrete Mathematics. North-Holland, 1986, vol. 29.

[66] L. Lovász and M. Plummer, *Matching Theory*. American Mathematical Society, Aug. 2009. [Online]. Available: http://dx.doi.org/10.1090/chel/367

[67] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, ser. Algorithms and Combinatorics. Springer, 2003, vol. 24.

[68] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora, "Structural reducibility of multilayer networks," *Nature Communications*, vol. 6, no. 1, Apr. 2015, publisher: Springer Science and Business Media LLC. [Online]. Available: http://dx.doi.org/10.1038/ncomms7864

[69] L. Zdeborová and F. Krzakala, "Statistical physics of inference: thresholds and algorithms," *Advances in Physics*, vol. 65, no. 5, pp. 453–552, Aug. 2016, publisher: Informa UK Limited. [Online]. Available: http://dx.doi.org/10.1080/00018732.2016.1211393

[70] X. Zhang, C. Pan, and W. Zhang, "Control hubs of complex networks and a polynomial-time identification algorithm," 2022, _eprint: 2206.01188. [Online]. Available: https://arxiv.org/abs/2206.01188

[71] P. ERDdS and A. R&wi, "On Random Graphs I," *Publ. math. debrecen*, vol. 6, no. 290-297, p. 18, 1959.

[72] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999. [Online]. Available: https://www.science.org/doi/10.1126/science.286.5439.509

[73] L. Perron and V. Furnon, "OR-Tools CP-SAT solver," Google OR-Tools documentation, 2023, https://developers.google.com/optimization.

[74] A. Hagberg, D. Schult, and P. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy)*, 2008, pp. 11–15.

[75] S. Mitchell, "Pulp: A linear programming toolkit for python," *The University of Auckland, Department of Engineering Science*, 2011, https://coin-or.github.io/pulp/.

[76] T. Ito, E. D. Demaine, N. J. A. Harvey, C. H. Papadimitriou, M. Sideri, R. Uehara, and Y. Uno, "On the complexity of reconfiguration problems," *Theoretical Computer Science*, vol. 412, no. 12–14, pp. 1054–1065, 2011.