Towards Automated Compliance Verification

Tamara Drucks[1,2], Patrick Indri[1], Sebastian Heil[3], and Martin Gaedke[3]

[1] Research Unit Machine Learning, TU Wien, Austria
[2] Research Group Data Mining and Machine Learning, University of Vienna, Austria
[3] Distributed and Self-Organizing Systems, TU Chemnitz, Germany

Abstract. We propose an algorithmic verification scheme to test the compliance of machine learning (ML) systems to regulatory demands. The de facto standard for testing the capabilities of large-scale ML systems, including compliance, is to use curated benchmark datasets. We argue that this is not sufficient and explore how policy requirements informed by regulatory specifications can be mapped to *quantifiable* machine learning (ML) properties, such as, e.g., differential privacy. We find that, often, it is both theoretically and practically not possible for an ML model to satisfy multiple policy requirements at once, as they are in conflict. We formalize this and show that it is generally intractable to find an ideal mapping from policy requirements to quantifiable ML properties. Next, we propose a novel compliance scheme in which a greedy algorithm takes into account conflicts between ML properties and outputs a *set* of ML systems that jointly satisfy the policy requirements. Finally, we implement our proposed compliance scheme and showcase its applicability for a selected set of policy requirements and ML properties.

Keywords: Compliance. · Trustworthy machine learning · Regulation.

1 Introduction

Recent policy and regulatory efforts [11,13,12] increasingly require machine learning (ML) models to meet desiderata such as robustness, fairness, and privacy. These desiderata are often expressed as high-level requirements that ML models are expected to satisfy in practice [32,38]. The de facto standard to test compliance of modern large-scale ML systems is to use curated benchmark datasets [46,21,19,40,9]. We argue that this is not sufficient and can only ever expose violations of regulatory demands, but not ensure compliance thereof. Another central difficulty in this process is that policy requirements do not directly translate to quantifiable ML properties. In fact, some policy requirements may be practically understood with different technical interpretations [17], while others may not have an obvious technical counterpart. Moreover, ML properties that can be enforced in isolation may be infeasible to implement in combination [37,36,15,25,34]. Motivated by these observations, our investigation is organized around the following question: *How can we ensure that deployed AI systems verifiably satisfy regulatory demands?* Regulatory initiatives such as the EU AI Act [12] impose abstract, and at times competing, obligations on deployed ML

systems, increasing the need for operational compliance methods. Benchmark-led audits, while pragmatic, identify violations only after the training process and cannot ensure compliance by construction. Because some requirements inherently conflict, effective oversight should document these tensions, prioritize trade-offs, and, if necessary, require multiple ML systems that jointly cover obligations. To address this, we propose to check for compliance in two ways: *a priori*, by guaranteeing certain ML properties that are inherent to the training process, and *posthoc*, by using held-out data to check for violations. We (*i*) formalize the problem of mapping quantifiable ML properties to a given set of policy requirements and show that this is generally intractable (see Proposition 1). Next, we (*ii*) propose a greedy, conflict-aware compliance scheme that returns a set of trained ML models that jointly satisfy the policy requirements in polynomial time (see Algorithm 1) and implement it for a selected set of policy requirements and ML properties. By making conflicts between regulatory demands and ML properties explicit and verifying compliance a priori, our approach supports accountable deployment aligned with public-interest requirements.

2 Related Work

There is a growing body of work that attempts to translate policy requirements for ML systems into concrete and actionable compliance measures that can be verified and documented. One line of research maps legal requirements to theoretical guarantees such as differential privacy or robustness [38,28,32], while another develops domain-specific languages and ontologies to make regulations machine-readable in the first place [18,29]. The dominant practical approach, however, is benchmarking: auditing ML models against curated test suites to check whether they satisfy regulatory specifications [19,2,6,16,33,31,3]. Benchmarks have become the standard approach because regulations are difficult for AI practitioners to interpret directly [48,24], and automated compliance checks offer a pragmatic shortcut. Benchmarking can, however, never enforce compliance by construction, but only identify violations of legal standards *posthoc*. Additionally, policy requirements in benchmarks are commonly treated in isolation, which assumes that all requirements can be implemented and satisfied independently. This assumption does not always hold, as ML properties that address policy requirements are often in tension with one another: The right to an explanation and the right to privacy, for instance, have been shown to conflict [37,36,30], and similar trade-offs arise between ML properties that ensure fairness, robustness or accuracy [25,34,14,5,15,22]. Even individual legal concepts do not always translate cleanly: Frohnapfel et al. [17] argue that measuring average accuracy alone is not compliant with current legislation and advocate to use predictive multiplicity instead. Taken together, these results show that some combinations of regulatory requirements are infeasible to satisfy simultaneously. This motivates our proposed shift from posthoc auditing toward building compliance guarantees directly into the training process, which forces conflicting requirements to be resolved by design rather than discovered after deployment.

3 Preliminaries

We briefly introduce some terminology and notation necessary for the remainder of the paper. We assume some familiarity with concepts from theoretical computer science including logics and basic graph theory; we refer readers unfamiliar with such concepts to Dasgupta et al. [8]. A *graph* $G = (V, E)$ consists of vertex set V and edge set E such that $uv = vu \in E$ if $u, v \in V$ are adjacent. A *bipartite graph* is a graph with $V = V_1 \cup V_2$ such that $V_1 \cap V_2 = \emptyset$ and $E \subseteq \{uv \mid u \in V_1, v \in V_2\}$. We denote the neighbors of a vertex v by $N(v) = \{u \mid uv \in E\}$ and its *degree* as $\deg(v) = |N(v)|$. We denote a subgraph H of G induced by vertex set $V' \subseteq V$ as $H = G[V']$. An *independent set* in a graph is a set of vertices such that each pair of vertices is non-adjacent. A *propositional logic formula* consists of variables x together with the operators conjunction $\wedge$, disjunction $\vee$ and negation $\neg$. A *literal* is a variable x or its negation $\neg x$. A *clause* is a disjunction of literals. Given a propositional logic formula φ in conjunctive normal form, e.g., $\varphi = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$, the satisfiability (SAT) problem asks whether there exists an assignment of truth values to variables in φ such that the entire formula φ evaluates to true.

4 From Policy Requirements to Trustworthy Models

In this section, we formalize the problem of jointly satisfying potentially conflicting regulatory requirements as a constrained learning problem. We first (*i*) introduce a conceptual mapping between a set of policy requirements to quantifiable ML properties that explicitly represents potential conflicts. Next, we (*ii*) formalize the task of finding an *ideal* mapping between policies and properties as the compliance decision problem Be Cheap But Compliant, and show that it is computationally intractable. This motivates us to (*iii*) propose efficient greedy algorithms that return *sets* of models satisfying prioritized policy requirements under a cost budget, which we (*iv*) empirically evaluate for a subset of selected policy requirements and machine learning properties.

4.1 Representing Requirements, Properties, and Conflicts

We formalize the task of mapping policy requirements to quantifiable machine learning properties guided by the following question: *How can we satisfy as many policy requirements as possible, while also ensuring technical feasibility?* We denote the policy requirements extracted from legal texts as $\mathcal{P}$ and the set of corresponding machine learning properties as $\mathcal{M}$. There exists a many-to-many mapping from $\mathcal{P}$ to $\mathcal{M}$, i.e., policy requirements might be satisfiable by more than one machine learning property, and the same machine learning property might address more than one policy requirement. For instance, the ML property *differential privacy* [10] formally protects individual-level privacy, but also makes ML models more adversarially robust [27]. The policy requirement *private*, on the other hand, can also be addressed by *k-anonymity* [41], another ML

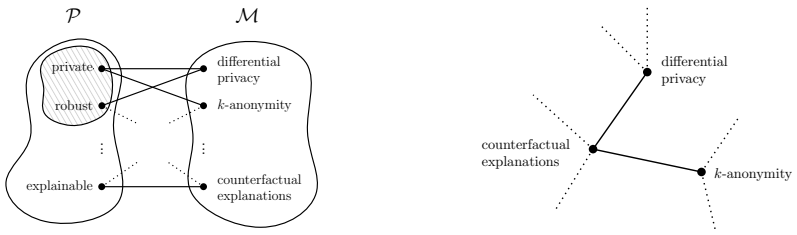

Fig. 1: *Left:* Policy-property graph G. The policy requirements in the shaded region are in P_{hard}. *Right:* Conflict graph G_C for policy-property graph G.

property that enforces user-level privacy. More formally, we model this many-to-many mapping as the relation $\mathcal{R} : \mathcal{P} \times \mathcal{M} \to \{0, 1\}$ such that $\mathcal{R}(p, m) = 1$ if policy requirement p is addressed by ML property m. We next construct the bipartite graph $G = (\mathcal{P} \cup \mathcal{M}, \{pm \mid p \in \mathcal{P}, m \in \mathcal{M}, \mathcal{R}(p, m) = 1\})$ by inserting an edge whenever policy requirement p is addressed by property m. Consider Figure 1 for an illustration of G: as the ML properties *differential privacy* and *k-anonymity* both address the policy requirements *private* and *robust*, we have edges *private-differential privacy*, *private-k-anonymity*, *robust-differential privacy* and *robust-k-anonymity*. Since enforcing certain ML properties can be costly – e.g., the standard implementation to ensure differential privacy is computationally expensive – each ML property $m \in \mathcal{M}$ is additionally associated with some cost $c(m)$; we remark that these costs do not have to directly correspond to technical costs, but might also stem from organizational overhead. Similarly, policymakers might regard some policies as more crucial than others, which we model in two ways: each policy requirement $p \in \mathcal{P}$ has some associated importance score $s(p)$ that helps decision-makers prioritize; and, in addition, we define some subset $P_{hard} \subseteq \mathcal{P}$ of *hard* policies that *must* be satisfied. For instance, we might consider the requirement that AI systems are *explainable* as non-negotiable, but allow some systems to be non-private if used in low-risk contexts or non-sensitive domains. To take into account the costs associated to certain machine learning properties as well as the importance scores associated with certain policy requirements, we set thresholds δ_c and δ_s, i.e., our costs are not allowed to exceed δ_c and our total importance score needs to reach at least δ_s. Finally, we represent pairwise conflicts between ML properties (m, m') in $\mathcal{M} \times \mathcal{M}$ in the conflict graph $G_C = (\mathcal{M}, \{mm' \mid m, m' \in \mathcal{M}, m \text{ and } m' \text{ are in conflict}\})$, where all conflicting ML properties are connected via an edge; refer to Figure 1, where G_C represents the discussed tensions between the right to explanation and the right to privacy.

Given the formalization of policy-property relationships, conflicts, importance, and costs, we next pose a compliance decision problem that asks for a low-cost, conflict-free selection covering required policies; we show it is intractable and then introduce greedy algorithms to navigate these trade-offs.

4.2 Why Exact Compliance Selection Is Hard

We next formalize the task of finding an ideal mapping as the decision problem Be Cheap But Compliant (BCBC) and show that BCBC is intractable.

Definition 1 (Be Cheap But Compliant (BCBC)). *Given policy-property graph G, conflict graph G_C, hard policy requirements $P_{hard} \subseteq \mathcal{P}$, importance scores $s : \mathcal{P} \to \mathbb{N}$, costs $c : \mathcal{M} \to \mathbb{R}$, cost threshold $\delta_c \geq 0$, and importance threshold $\delta_s \geq 0$, the* Be Cheap But Compliant *problem is defined as follows. Does there exist a set of ML properties $M_{ass} \subseteq \mathcal{M}$ such that the following holds:*

1. *M_{ass} is an* independent set *in G_C;*
2. *$P_{hard} \subseteq P_{ass}$, where $P_{ass} = \bigcup_{m \in M_{ass}} N_G(m)$;*
3. *$\sum_{p \in P_{ass}} s(p) \geq \delta_s$;*
4. *$\sum_{m \in M_{ass}} c(m) \leq \delta_c$.*

In other words, Be Cheap But Compliant asks whether there is a set of ML properties that together address all *hard* policy requirements, are not in conflict and do not exceed the cost score while reaching the importance score threshold with the satisfied policy requirements. Unfortunately, finding an ideal mapping from policy requirements to ML properties is NP-complete, which we state more formally in Proposition 1.

Proposition 1. *The Be Cheap But Compliant problem is NP-complete.*

Proof sketch. We prove Proposition 1 by a reduction from SAT. The main idea of the reduction is to map literals to ML properties and clauses to policy requirements and thus coverage constraints. Please find the full proof in Section A in the Appendix.

Proposition 1 implies that as the number of regulatory requirements and potential conflicts grows, finding an exact, single best selection of technical guarantees can quickly become computationally prohibitive in practice. Therefore, we next propose a new compliance scheme that returns a *set* of ML models that jointly satisfy a given set of required policy demands. We realize this via two polynomial-time greedy algorithms that favor either broad coverage or low-conflict solutions under the cost threshold, making explicit when multiple models are needed to meet prioritized requirements.

4.3 An Efficient Algorithm for Conflict-Aware Compliance

We now propose two variants of an efficient greedy algorithm that returns a *set* of conflict-free ML models that jointly satisfy the policy requirements. We assume that properties selected to cover hard requirements are always included, irrespective of δ_c; their combined cost forms a mandatory baseline, and δ_c bounds only the additional cost incurred for soft requirements. For tractability, we prioritize high-importance requirements by processing them in decreasing order of $s(p)$, without guaranteeing that the importance threshold δ_s is reached. Given

a family of ML models $\mathcal{H}$, policy-property graph G, and conflict graph G_C, our greedy algorithm outputs a *set of trained ML models* $H \subseteq \mathcal{H}$ that aims to jointly satisfy all requirements. We introduce two variants: (*i*) GreedyMaxDegree in (see Algorithm 1), which chooses high-degree vertices from $\mathcal{M}$ in G and therefore aims to cover many policy requirements in each iteration, and (*ii*) GreedyMinDegree (see Remark 1), which chooses low-degree vertices in G_C and therefore aims to reduce the number of conflicts among selected properties.

Algorithm 1 GreedyMaxDegree

Input: Set of policy requirements $\mathcal{P}$, set of ML properties $\mathcal{M}$, policy-property graph G, conflict graph G_C, hard policy requirements P_{hard}, importance score function s, cost function c, cost threshold δ_c, family of ML models $\mathcal{H}$
Output: Set of trained ML models H

$H \leftarrow \emptyset$
$M_{\text{sel}} \leftarrow \emptyset$
$C \leftarrow \emptyset$
Sort $\mathcal{P}$ in decreasing $s(p)$
for each policy requirement p_i in this order **do**
  $m_i \leftarrow \arg\max_{m \in N_G(p_i)} \deg_G(m)$
  if $m_i \in M_{\text{sel}}$ **then**
    continue
  end if
  if $p_i \notin P_{hard}$ **then**
    $C \leftarrow C + c(m_i)$
    if $C > \delta_c$ **then**
      $C \leftarrow C - c(m_i)$
      continue
    end if
  end if
  if there exists $M_j \in M_{\text{sel}}$ such that $M_j \cup \{m_i\}$ is an independent set in G_C **then**
    $M_j \leftarrow M_j \cup \{m_i\}$
  else
    $M_{\text{sel}} \leftarrow M_{\text{sel}} \cup \{\{m_i\}\}$
  end if
end for
for each $M_j \in M_{\text{sel}}$ **do**
  Train one $h_j \in \mathcal{H}$ satisfying all ML properties in M_j
  $H \leftarrow H \cup \{h_j\}$
end for
return H

Remark 1. The GreedyMinDegree variant is obtained from Algorithm 1 by replacing the selection rule $m_i \leftarrow \arg\max_{m \in N_G(p_i)} \deg_G(m)$ in line 7 with the alternative rule $m_i \leftarrow \arg\min_{m \in N_G(p_i)} \deg_{G_C}(m)$.

GreedyMaxDegree selects ML properties that cover most policy requirements at once, maximizing coverage within a given cost budget; this is useful when the goal is broad, rapid progress across requirements, as in early-stage audit preparation. The trade-off is that high-coverage properties tend to conflict with one another, making it more likely that some requirements cannot be jointly satisfied and that trade-offs must be explicitly documented. GreedyMinDegree instead prioritizes properties with few known conflicts, yielding a more coherent and justifiable set of guarantees. This is better suited for high-stakes or rights-critical deployments where regulators expect a clear rationale for each design choice, even if fewer requirements are covered overall. GreedyMaxDegree and GreedyMinDegree both ensure that all hard policy requirements are satisfied and aim to maximize the importance scores while not exceeding the cost threshold. Both algorithms run in polynomial time (see Section B in the Appendix for details), meaning compliance can be ensured efficiently even as the number of policy requirements grows. This stands in stark contrast to the underlying selection problem, which is NP-complete, i.e., finding the provably optimal selection of ML properties would become computationally infeasible as regulations expand. The greedy approach thus offers a practical path to compliance planning at scale, without sacrificing tractability.

4.4 Verifiably Compliant Models in Practice

In this section, we implement our algorithmic verification scheme to evaluate its utility for a selected set of properties and requirements. To find a good mapping from policy requirements to ML properties, we implement the greedy algorithms presented in the previous section. Our scheme then consists of two stages for each model: (*i*) a training process in which the chosen ML properties provide *a priori* guarantees and (*ii*) *posthoc* checks for compliance violation on held-out test data. In the following, we will briefly describe our input data, the experimental setup, and present results for the popular vision dataset CIFAR10 [26]. Please find more details on the experimental setup in Section C and additional results on Fashion-MNIST [44] in Section D in the Appendix.

Policy-property graph and conflict graph. To build a BCBC input instance, we choose the policy requirements listed in Table 1; note that the content of this table is exemplary and not an exhaustive list. We thus set $\mathcal{P} = \{$explainable, transparent, robust, private$\}$ and $\mathcal{M} = \{$counterfactual explanations (CF), conformal prediction (CP), differential privacy (DP)$\}$. In our scenario, explainability and transparency are prioritized with importance scores of $s(\text{explainable}) = 10$ and $s(\text{transparent}) = 8$, followed by $s(\text{robust}) = 6$ and $s(\text{private}) = 4$, and we omit hard policy requirements and costs. The property-policy graph is constructed similarly to Figure 1 (*left*): The policy requirement *explainable* is addressed by ML property CF and *transparent* is addressed by CP. The requirement *robust* is addressed by both, CF as well as DP, as the former is commonly trained with

adversarial robustness methods to provide more stable explanations[4] [39,42]. Lastly, the policy requirement *private* is addressed by ML property DP. The conflict graph consists of the single edge DP−CF. In this setting, both MaxDegree as well as MinDegree are expected to return the same result: an *explainable*, *transparent* and *robust* model H_1 that implements CF and CP, and a *private* model H_2 that implements DP.

Table 1: Exemplary mapping from EU AI Act requirements to ML properties.

Policy requirement	Source	ML property
transparency	EU AI Act Art. 13(1)–(2)	Conformal prediction sets (coverage guarantees); calibrated probabilities [20,4]
explainability	EU AI Act Art. 13(3)(b)(iv)	Counterfactual explanations [43,23]
robustness	EU AI Act Art. 15	Certified adversarial robustness (randomized smoothing; differential privacy) [7,27]
privacy	EU AI Act Art. 10(5)	Differential privacy (e.g., DP-SGD); k-anonymity [10,1,41]

Experimental setup. We test our compliance scheme on the two vision datasets CIFAR10 [26] and Fashion-MNIST [44]. In addition to Algorithm 1, which returns a set of trained models, we also train two baseline models. We perform three runs for each dataset, our compliance scheme, and each baseline, and report results averaged across the runs. We ensure *a priori* compliance in the training process and check for empirical compliance via a *posthoc* audit on the trained models. We consider the latter to be successful, if the models pass predefined threshold values; for our compliance scheme, we perform the posthoc compliance checks only for the model that addresses the tested policy requirement. Please find more details on the experimental setup below as well as in Section C in the Appendix. *Baselines.* We compare our compliance scheme with two baselines: NG-Baseline is a model trained with no theoretical guarantees, while All-Baseline is a model that is trained with all ML properties at once, ignoring potential conflicts. *Model cards.* In addition to the guaranteed properties, our algorithm also returns model cards that detail the training procedure and how each policy requirement is satisfied. The model cards are in the human-readable and machine-checkable format of JSON files. *A priori compliance.* For verifiably *private* models, we use DP-SGD [1] with $\epsilon = 5$ and $\delta = 10^{-5}$. For

[4] Note that TRADES [47], which we use, does not provide per-example *certified* robustness, but provides formal guarantees on the robust error on a population level; we leave the implementation of per-example *certified* robustness for future work.

transparent models, we use split conformal prediction [4] with a target coverage of 90% as well as model cards. For *explainable* models, we train the models with the TRADES loss [47] to obtain more stable explanations which we compute via projected gradient descent. *Robustness* is thus implicitly implemented in CF, as well as through DP [27]. *Posthoc compliance.* To assess whether the trained models are violating any of the required policies, we perform additional checks on held-out test data. For privacy, we conduct membership inference attacks; for transparency, we compute the expected calibration error as well as the empirical coverage; for explainability, we compute the average L_2 distance to the counterfactual explanations; for robustness, we compute the drop in average accuracy for adversarial input samples. To assess compliance, we choose reasonable threshold values detailed in Section C in the Appendix.

Results. As expected, both MaxDegree as well as MinDegree return the same mapping: one model where CF addresses explainability as well as robustness and CP addresses transparency, and one model where DP addresses privacy. We present the results for running MaxDegree on CIFAR10 in the table below; additional results on Fashion-MNIST can be found in Section D in the Appendix. Table 2 shows that, overall, our verifiable compliance scheme per-

Table 2: Compliance checks for CIFAR10

Model	Guarantees	Transparent	Explainable	Robust	Private
NG-Baseline	✗	✗	✓	✗	✗
All-Baseline	✓	✗	✗	✗	✓
Ours	✓	✓	✓	✗	✓

forms best. NG-Baseline, which offers no theoretical guarantees, only passes the compliance checks for explainability, and fails for the remaining policy requirements. All-Baseline, on the other hand, provides formal guarantees, but fails for all policy requirements except for *privacy*. Our set-based approach provides formal guarantees and only fails for the robustness check. This could be due to the TRADES loss bounding the robust error at the population level without per-example certificates. Alternative approaches such as randomized smoothing [7] could provide per-example certificates, at a higher computational cost. We point out that All-Baseline and our model still achieve an accuracy of around 20-25%, while the NG-Baseline has an accuracy of 0%; thus, while both models with guarantees fail the compliance check for the threshold of 40% accuracy, they perform significantly better than the model without any guarantees.

5 Conclusion & Outlook

We proposed a two-stage compliance scheme for ML systems combining *a priori* guarantees with *posthoc* compliance violation checks, and showed that the underlying selection problem is NP-complete. We provided a tractable solution by introducing two efficient greedy algorithms that return sets of ML models that aim to jointly satisfy prioritized policy requirements while making conflicts explicit. We instantiated our proposed compliance scheme on concrete policy requirements and ML properties to evaluate the greedy algorithms empirically, and integrated model cards as a transparency mechanism to better support regulatory oversight. For future work, we will provide approximation guarantees for the greedy algorithms and perform an exhaustive empirical evaluation.

Implications for Policy Makers. Since some policy goals cannot be satisfied simultaneously, effective oversight should require explicit conflict documentation, prioritized trade-offs, and, where necessary, multiple models jointly covering requirements rather than a single all-in-one system. Implementing these practices would improve accountability and auditability for regulatory oversight in democratic contexts.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. pp. 308–318 (2016)
2. Alamdari, P.A., Klassen, T.Q., McIlraith, S.A.: Auditing, monitoring, and intervention for compliance of advanced ai systems. In: ICML 2025 Workshop on Reliable and Responsible Foundation Models (2025)
3. Ameur, M., Brik, B., Ksentini, A.: Eurocomply: Enabling zero-touch ai compliance auditing via llm-based agentic ai. IEEE Communications Magazine (2026)
4. Angelopoulos, A.N., Bates, S.: A gentle introduction to conformal prediction and distribution-free uncertainty quantification (2022)
5. Bell, A., Solano-Kamaiko, I., Nov, O., Stoyanovich, J.: It's just not that simple: An empirical study of the accuracy-explainability trade-off in machine learning for public policy. In: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency. p. 248–266. FAccT '22, Association for Computing Machinery, New York, NY, USA (2022)
6. Bourrée, J.G., Godinot, A., Biswas, S., Kermarrec, A.M., Le Merrer, E., Tredan, G., De Vos, M., Vujasinovic, M.: Robust ml auditing using prior knowledge. In: International Conference on Machine Learning. pp. 18794–18810. PMLR (2025)
7. Cohen, J., Rosenfeld, E., Kolter, Z.: Certified adversarial robustness via randomized smoothing. In: international conference on machine learning. pp. 1310–1320. PMLR (2019)
8. Dasgupta, S., Papadimitriou, C.H., Vazirani, U.V.: Algorithms. McGraw-Hill Higher Education, New York (2008)

9. Davvetas, A., Papademas, M., Ziouvelou, X., Karkaletsis, V.: Ai act evaluation benchmark: An open, transparent, and reproducible evaluation dataset for nlp and rag systems. arXiv preprint arXiv:2603.09435 (2026)
10. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of cryptography conference. pp. 265–284. Springer (2006)
11. European Parliament, Council of the European Union: Regulation (eu) 2022/2065 of the european parliament and of the council of 19 october 2022 on a single market for digital services and amending directive 2000/31/ec (digital services act). Official Journal of the European Union **L 277**, 1–102 (oct 2022)
12. European Parliament, Council of the European Union: Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence and amending regulations (ec) no 300/2008, (eu) no 167/2013, (eu) no 168/2013, (eu) 2018/858, (eu) 2018/1139 and (eu) 2019/2144 and directive 2014/90/eu, (eu) 2016/797 and (eu) 2018/2001 (artificial intelligence act). Official Journal of the European Union **L 2024/1689** (2024)
13. European Parliament and Council: Regulation (EU) 2023/2854 of the European Parliament and of the Council of 13 December 2023 on harmonised rules on fair access to and use of data and amending Regulation (EU) 2017/2394 and Directive (EU) 2020/1828 (Data Act) (2023)
14. Ferry, J., Aïvodji, U., Gambs, S., Huguet, M.J., Siala, M.: Taming the triangle: On the interplays between fairness, interpretability, and privacy in machine learning. Computational Intelligence **41**(4), e70113 (2025)
15. Fokkema, H., De Heide, R., Van Erven, T.: Attribution-based explanations that provide recourse cannot be robust. Journal of Machine Learning Research **24**(360), 1–37 (2023)
16. Friedrich, F., Brack, M., Struppek, L., Hintersdorf, D., Schramowski, P., Luccioni, S., Kersting, K.: Auditing and instructing text-to-image generation models on fairness. AI and Ethics **5**(3), 2103–2123 (2025)
17. Frohnapfel, K., Seyfert, M., Bordt, S., von Luxburg, U., Meding, K.: Using predictive multiplicity to measure individual performance within the ai act. arXiv preprint arXiv:2602.11944 (2026)
18. Grünewald, E., Pallas, F.: Tilt: A gdpr-aligned transparency information language and toolkit for practical privacy engineering. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. pp. 636–646 (2021)
19. Guldimann, P., Spiridonov, A., Staab, R., Jovanovi'c, N., Vero, M., Vechev, V., Gueorguieva, A., Balunovi'c, M., Konstantinov, N., Bielik, P., Tsankov, P., Vechev, M.T.: Compl-ai framework: A technical interpretation and llm benchmarking suite for the eu artificial intelligence act. ArXiv **abs/2410.07959** (2024)
20. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International conference on machine learning. pp. 1321–1330. PMLR (2017)
21. Hu, W., Jing, H., Shi, H., Li, H., Song, Y.: Safety compliance: Rethinking llm safety reasoning through the lens of compliance (2025)
22. Indri, P., Drucks, T., Gärtner, T.: On the trade-off between expressivity and privacy in graph representation learning. In: The Fourteenth International Conference on Learning Representations (2026)
23. Karimi, A.H., Barthe, G., Balle, B., Valera, I.: Model-agnostic counterfactual explanations for consequential decisions. In: International conference on artificial intelligence and statistics. pp. 895–905. PMLR (2020)

24. Kelly, J., Zafar, S.A., Heidemann, L., Zacchi, J.V., Espinoza, D., Mata, N.: Navigating the eu ai act: A methodological approach to compliance for safety-critical products. In: 2024 IEEE Conference on Artificial Intelligence (CAI). pp. 979–984. IEEE (2024)
25. Krishna, S., Ma, J., Lakkaraju, H.: Towards bridging the gaps between the right to explanation and the right to be forgotten. In: International Conference on Machine Learning. pp. 17808–17826. PMLR (2023)
26. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
27. Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., Jana, S.: Certified robustness to adversarial examples with differential privacy. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 656–672. IEEE Computer Society (2019)
28. Lewis, D., Lasek-Markey, M., Golpayegani, D., Pandit, H.J.: Mapping the regulatory learning space for the eu ai act. arXiv preprint arXiv:2503.05787 (2025)
29. Leyva-Sánchez, S., Linde, F., Manab, M.A., Poveda-Villalón, M., Rodríguez-Doncel, V.: Daont: A formal ontology for eu data act compliance. arXiv preprint arXiv:2604.16386 (2026)
30. Manna, S., Sett, N.: Reconciling privacy and explainability in high-stakes: A systematic inquiry (2025)
31. Nguyen, L.P.T., Do, H.T.: Raise: A unified framework for responsible ai scoring and evaluation. In: International Conference on Principles and Practice of Multi-Agent Systems. pp. 453–460. Springer (2025)
32. Nolte, H., Rateike, M., Finck, M.: Robustness and cybersecurity in the eu artificial intelligence act. In: Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency. pp. 283–295 (2025)
33. Obradov, A., Pavković, B.: Ensuring compliance of high-risk ai systems with the eu ai act using open source libraries. In: 2025 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC). pp. 54–59. IEEE (2025)
34. Oesterling, A., Ma, J., Calmon, F., Lakkaraju, H.: Fair machine unlearning: Data removal while mitigating disparities. In: Dasgupta, S., Mandt, S., Li, Y. (eds.) Proceedings of The 27th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 238, pp. 3736–3744. PMLR (02–04 May 2024)
35. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
36. Pawelczyk, M., Lakkaraju, H., Neel, S.: On the privacy risks of algorithmic recourse. In: International Conference on Artificial Intelligence and Statistics. pp. 9680–9696. PMLR (2023)
37. Pawelczyk, M., Leemann, T., Biega, A., Kasneci, G.: On the trade-off between actionable explanations and the right to be forgotten. arXiv preprint arXiv:2208.14137 (2022)
38. Raza, S., Sapkota, R., Karkee, M., Emmanouilidis, C.: Trism for agentic ai: A review of trust, risk, and security management in llm-based agentic multi-agent systems. AI Open **7**, 71–95 (2026)
39. Slack, D., Hilgard, A., Lakkaraju, H., Singh, S.: Counterfactual explanations can be manipulated. Advances in neural information processing systems **34**, 62–75 (2021)
40. Song, D., Huang, Y., Chen, B., Cong, T., Goebel, R., Ma, L., Khomh, F.: Evaluating implicit regulatory compliance in llm tool invocation via logic-guided synthesis. arXiv preprint arXiv:2601.08196 (2026)

41. Sweeney, L.: k-anonymity: A model for protecting privacy. International journal of uncertainty, fuzziness and knowledge-based systems **10**(05), 557–570 (2002)
42. Upadhyay, S., Joshi, S., Lakkaraju, H.: Towards robust and reliable algorithmic recourse. Advances in Neural Information Processing Systems **34**, 16926–16937 (2021)
43. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the gdpr. SSRN Electronic Journal **31**(2), 841–887 (2017)
44. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
45. Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., Cormode, G., Mironov, I.: Opacus: User-friendly differential privacy library in PyTorch. arXiv preprint arXiv:2109.12298 (2021)
46. Zeng, Y., Yang, Y., Zhou, A., Tan, J., Tu, Y., Mai, Y., Klyman, K., Pan, M., Jia, R., Song, D., et al.: Air-bench 2024: A safety benchmark based on regulation and policies specified risk categories. In: International Conference on Learning Representations. vol. 2025, pp. 63997–64031 (2025)
47. Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: International conference on machine learning. pp. 7472–7482. PMLR (2019)
48. Zhang, S., Maharjan, S.: Security, privacy, and agentic ai in a regulatory view: From definitions and distinctions to provisions and reflections (2026)

A Missing Proof

In this section, we provide the missing proof for Proposition 1.

Proposition 1. *The Be Cheap But Compliant problem is NP-complete.*

Proof. For the proof, we consider the special case of $c(m) = 0$ for all $m \in \mathcal{M}$, $P_{hard} = \mathcal{P}$ and $s(p) = \infty$ for all $p \in \mathcal{P}$. BCBC is in NP: Given a subgraph $H = G[P_{ass} \cup M_{ass}]$ we can check in $\mathcal{O}(n)$ time whether $\deg(p) \geq 1$ for $|\mathcal{P}| = n$ and in $\mathcal{O}(k^2)$ time whether M_{ass} is an independent set in G_C for $|\mathcal{M}| = k$. What remains to show is that BCBC is NP-hard, which we do by a reduction from SAT. We first (i) define a mapping from SAT to BCBC, and then (ii) – (iii) show correctness in both directions. (i) Mapping SAT instance to BCBC instance: Given our SAT instance φ with m variables and n clauses in CNF, we construct the incidence graph $G_\varphi = (C \cup L, \{uv \mid u \in C, v \in L, \text{literal } v \text{ occurs in clause } u\})$ of SAT, where node set C contains all clauses, and node set L contains all literals (positive and negative). Additionally, we define a conflict graph $G_{\text{var}(\varphi)} = (L, \{uv \mid u, v \in L, u = \neg v\})$ that consists of all literals as nodes and an edge for every pair of positive and negative literals of the same variable. We reduce our SAT instance $(G_\varphi, G_{\text{var}(\varphi)})$ to a BCBC (G, G_C) instance as follows. We first construct G and G_C by setting $C = \mathcal{P}$ and $L = \mathcal{M}$. The construction is in polynomial time, as creating the incidence graph takes $\mathcal{O}(n+k)$ time, and creating the conflict graph takes $\mathcal{O}(k)$ time. Constructing G from G_φ and G_C from $G_{\text{var}(\varphi)}$ takes constant time. (ii) SAT-Yes-instance $\implies$ BCBC-Yes-instance:

Following our construction, a satisfying assignment for $(G_\varphi, G_{\text{var}(\varphi)})$ is defined as a subgraph $H = G_\varphi[C_{ass} \cup L_{ass}]$ such that $\forall c \in C_{ass} \subseteq C$, $\deg(c) \geq 1$ and $L_{ass} \subseteq L$ forms an independent set in $G_{\text{var}(\varphi)}$. Applying our mapping $C = \mathcal{P}, L = \mathcal{M}$, we obtain $H' = G[P_{ass} \cup M_{ass}]$ such that $\forall p \in P_{ass} \subseteq \mathcal{P}$, $\deg(p) \geq 1$ and $M_{ass} \subseteq \mathcal{M}$ forms an independent set in G_C, which is, by definition, a Yes-instance of BCBC. (*iii*) BCBC-Yes-instance $\implies$ SAT-Yes-instance: H is a BCBC-Yes-instance if $\forall p \in P_{ass}$, $\deg(p) \geq 1$ and M_{ass} is an independent set in G_C. Considering our mapping, this corresponds to the SAT instance $H' = G_\varphi[C_{ass} \cup L_{ass}]$ such that $\forall c \in C_{ass} \subseteq C$, $\deg(c) \geq 1$ and $L_{ass} \subseteq L$ forms an independent set in $G_{\text{var}(\varphi)}$, which is a satisfying assignment of φ.

B Runtime Complexity of Algorithm 1

In this section, we briefly discuss the exact runtime of Algorithm 1.

The sorting step in line 6 takes $\mathcal{O}(n \log n)$ time, looping through the policy requirements in lines 7–16 takes $\mathcal{O}(n)$ time, and each selected property may be checked against all previously selected properties in G_C which is in $\mathcal{O}(n)$. As $\mathcal{M} = k \leq n$, the overall runtime of both algorithms is thus dominated by $\mathcal{O}(n^2)$ for n policy requirements.

C Experimental Setup

We implemented our prototype in Python and used PyTorch [35] for the training process and Opacus [45] for ensuring differential privacy. We conducted all experiments on a consumer-grade GPU. For both datasets, we set $\mathcal{H}$ to be convolutional neural networks with three convolutional layers, two fully-connected layers, and used ReLU as activation function. We use group norm for the convolutional layers and dropout of 0.3 for the fully-connected layers. For the non-DP training, we use the Adam optimizer, a learning rate of 10^{-3}, cross-entropy loss, batch size 256, and early stopping with a patience of 5 and minimum change of 10^{-3}. For the private training, we use DP-SGD as optimizer with a learning rate of 0.05, momentum term of 0.9, maximum gradient norm of 1 and a logical batch size of 1024. We train CIFAR10 for a maximum of 50 epochs, and Fashion-MNIST for a maximum of 30 epochs; for the private training the number of epochs equals the maximum number of epochs to ensure that the target ϵ value is reached. For the posthoc compliance checks we use the thresholds listed in Table 3.

D Additional Results

We present the results of the *posthoc* compliance checks for Fashion-MNIST in Table 4. On Fashion-MNIST our approach fails the explainability check by a small margin. One possible explanation is that the TRADES loss flattens the

Table 3: Compliance audit thresholds.

Policy	Metric	Threshold
P_{privacy}	Membership inference attack advantage	≤ 0.10
P_{robust}	Projected gradient descent (ϵ=0.1) adversarial accuracy	≥ 0.40
P_{explain}	Mean counterfactual L_2 distance	≤ 0.30
$P_{\text{transparent}}$	Mean per-class expected calibration error	≤ 0.10
$P_{\text{transparent}}$	Empirical conformal coverage	≥ 0.88

loss surface around training points, thus pushing the decision boundaries away from the training data and possibly making counterfactuals more distant. This shows that, despite often providing more stable explanations, robustness and explainability still exhibit a quantitative tension, which has to be further investigated. Additionally, we can observe how the NG-Baseline passed all checks but robustness despite carrying no guarantees. This supports, rather than undermines, our proposed two-stage compliance scheme: Fashion-MNIST is an easy benchmark dataset and a standard model is incidentally private and incidentally well-calibrated. Posthoc auditing alone cannot tell incidental compliance from guaranteed compliance. Incidental compliance also does not transfer, as a similar baseline approach performs significantly worse on CIFAR-10 (Table 2).

Table 4: Compliance checks for Fashion-MNIST

Model	Guarantees	Transparent	Explainable	Robust	Private
NG-Baseline	✗	✓	✓	✗	✓
All-Baseline	✓	✗	✗	✓	✓
Ours	✓	✓	✗	✓	✓